

What is Event Knowledge Graph: A Survey

Saiping Guan, Xueqi Cheng, Long Bai, Fujun Zhang, Zixuan Li, Yutao Zeng, Xiaolong Jin, and Jiafeng Guo

Abstract—Besides entity-centric knowledge, usually organized as Knowledge Graph (KG), events are also an essential kind of knowledge in the world, which trigger the spring up of event-centric knowledge representation form like Event KG (EKG). It plays an increasingly important role in many downstream applications, such as search, question-answering, recommendation, financial quantitative investments, and text generation. This paper provides a comprehensive survey of EKG from history, ontology, instance, and application views. Specifically, to characterize EKG thoroughly, we focus on its history, definitions, schema induction, acquisition, related representative graphs/systems, and applications. The development processes and trends are studied therein. We further summarize prospective directions to facilitate future research on EKG.

Index Terms—Event knowledge graph, schema, event acquisition, script event prediction, temporal knowledge graph prediction.

1 INTRODUCTION

KNOWLEDGE Graph (KG), announced by Google in 2012, is a popular knowledge representation form. It focuses on entities and their relations, thus representing static knowledge. However, there is a great deal of event information in the world, which conveys dynamic and procedural knowledge. Thus, event-centric knowledge representation form like Event KG (EKG) is also essential. It has facilitated many downstream applications, such as search, question-answering, recommendation, financial quantitative investments, and text generation [1], [2], [3], [4], [5], [6].

This paper goes deep into what EKG is and how it develops. What do you want to know about EKG? You may be interested in how it comes into being, what is called an EKG, how to construct it, and further where it can be applied. Thus, to comprehensively introduce EKG, we see it from history, ontology, instance, and application views. As illustrated in Fig. 1, ontology and instance parts cooperate to form EKG, and EKG further supports many applications. Detailedly, from the history view, we present the brief history and our derived definition of EKG (see Section 2). From the ontology view, we study the basic concepts related to EKG, and the tasks and methods therein, including event schema, script, and EKG schema inductions (see Section 3). From the instance view, we elaborate on event acquisition and EKG-related representative graphs/systems (see Section 4). From the application view, we introduce some basic and deep applications on EKG (see Section 5). Moreover, the development processes and trends of the related tasks

are studied thoroughly. Future directions are then pointed out in Section 6, followed by conclusions in Section 7.

There are also some surveys related to part of EKG, focusing on event extraction [7], [8], [9], [10], event modeling and mining [11], event and event relation extractions [12], and event coreference resolution [13]. However, there is a lack of a comprehensive and deep survey on EKG. Actually, events are essential and non-negligible in the world. Many events happen every day, reflecting the state of the world. Thus, it is necessary to go deep into events. In this way, a comprehensive survey on EKG is of great importance.

2 WHAT IS EKG: HISTORY VIEW

In this section, from the history view, we present the brief history of EKG. We then derive our definition of EKG based on EKG-related concepts from history.

2.1 The Brief History of EKG

EKG does not suddenly burst forth. Instead, it is the outcome of the development of Natural Language Processing (NLP) and artificial intelligence. As presented in Fig. 2, the history of EKG can be divided into four stages and dates from event, event extraction, event relation extraction, etc.

Stage 1: Early stage of event constitution study. From 1950s, events and their constituents were extensively studied [14], [15], [16], [17]. For example, Davidson [15] tried to get the logical form of sentences about actions. He gave an account of the logical or grammatical roles of the words in such sentences. Mourelatos [16] and Pustejovsky [17] explored event and proposed its basic definition. In 1978, Mourelatos [16] defined events as inherently countable occurrences. In 1991, Pustejovsky [17] regarded that events provide distinct representations for linguistic analysis involving the aspectual properties of verbs, adverbial scopes, argument roles, and mappings from the lexicon to syntax.

Stage 2: Standard formed for event element extraction and ordered event structures appeared. In 1989, the MUC (Message Understanding Conference) evaluations proposed

• Saiping Guan, Xueqi Cheng, Long Bai, Fujun Zhang, Zixuan Li, Xiaolong Jin, and Jiafeng Guo are with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100864, China. E-mail: {guansaiping, cxq, bailong18b, zhangfujun19s, lizixuan, jinxiaolong, guojiafeng}@ict.ac.cn.

• Yutao Zeng is with the Platform and Content Group, Tencent, Beijing, 100080, China. E-mail: tedzeng@tencent.com.

Manuscript received April 19, 2005; revised August 26, 2015 (Corresponding author: Xueqi Cheng, Xiaolong Jin, and Jiafeng Guo).

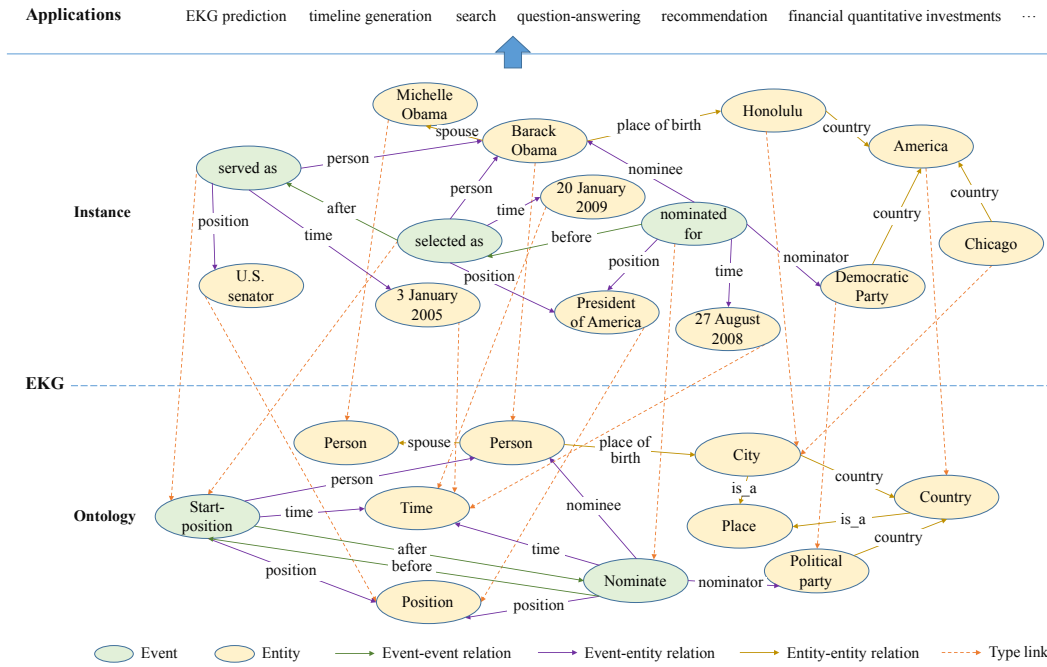


Fig. 1. A toy example of EKG and its applications.

event template filling, initiated by Naval Ocean System Center to foster the automatic analyses of military textual messages [18]. Given the description of events, participants were required to fill a template for each event on the page. More practically, with the unbounded information-bearing potential of the Web, the ACE (Automatic Content Extraction) program develops the capability to extract meaning therein. Starting in 2004, it added in event extraction, defined to extract event triggers and arguments, more in line with reality [19]. An event trigger is a word or span that most clearly expresses the event, i.e., indicates the event type, and an argument is an entity or span that plays a specific role in the event. With the awareness of the importance of identifying the events described in a text and locating them in time, in 2007, SemEval (Semantic Evaluation) proposed the temporal relation extraction task TempEval [20]. It extracts event temporal relations from texts. After that, researches on event and event relation extractions usually follow the task definitions of ACE and TempEval, respectively. Since understanding the temporal flow of discourse is significant for text comprehension, from 2006, there have been some attempts to construct ordered event structures from texts, such as temporal graph [21] and event timeline [22], [23].

Stage 3: KG and event graph appeared. Notably, in 2012, to enhance the results returned by Google searches, Google proposed KG with all the gathered knowledge of entities and relations in a semantic network. KG has thus caught much attention in various fields since then. However, it is about entities and their relations, i.e., static knowledge, and cannot deal with events elegantly. It somehow triggers the emergence of knowledge representation form on events and their relations. In 2014, Glavaš and Šnajder [24] proposed event graph that structures the information about events in texts to address the need for efficient retrieval and presentation of event-related information. In this event graph, nodes are events consisting of triggers and arguments (subject, object, time, and location), and edges are event temporal

relations. In 2015, Glavaš and Šnajder [25] further added in event coreference relations. To describe changes in the world, in 2016, Rospocher et al. [26] proposed event-centric KG, where nodes are events identified by URIs and entities, while edges are event-event relations, event-entity relations, and general facts about entities. The event-event relations include temporal and causal ones. Action, participant, time, and location are considered for the event-entity relations, capturing (what, who, when, and where).

Stage 4: Event logic graph came into being. Recently, with the development of many real-world applications, such as event prediction, decision making, and scenario design of dialog systems, there is a great need to understand the evolution and development of events. Thus, in 2017, Li et al. [27] proposed event evolutionary graph. It is similar to event graph [24], but its event nodes are abstract, generalized, and semantic complete verb phrases. It further considers event causal relations and reveals evolutionary patterns and development logics of events. Then, in 2018, Gottschalk and Demidova [28] proposed event-centric temporal KG, where events, entities, and relations are nodes, to facilitate semantic analyses regarding contemporary and historical events on the Web, in the news, and in social media. Events therein have topical, temporal, and geographical information, and are linked to the entities participating in the events. They also considered subevent, previous, and next event relations between events, and entity-entity relations. In 2019, event evolutionary graph was derived to event logic graph [29], where nodes are abstract, generalized, and semantic complete event tuples (s, p, o) , s is the actor/subject, p is the action/predicate (i.e., event trigger), and o is the object. Besides, two more event relations, conditional and hypernym-hyponym, were considered.

Generally speaking, there are many EKG-related concepts. As presented in Table 1, event evolutionary graph [27] and event logic graph [29] only focus on schema-level event knowledge. Nodes in event graph [24], [25] and event

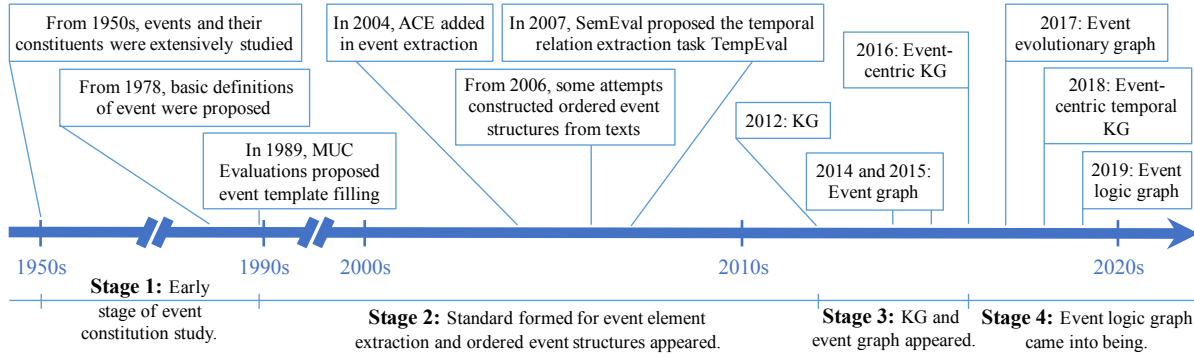


Fig. 2. The development history of EKG.

TABLE 1
Comparison of EKG-related concepts.

| Concept | Nodes | Edges | Features |
|--------------------------------|---|--|------------------|
| KG | Entities | Entity-entity relations | Static knowledge |
| Event graph [24], [25] | Events consisting of triggers and arguments (subject, object, time, and location) | Temporal relations between events [24]; Temporal and coreference relations between events [25] | Instance level |
| Event-centric KG [26] | Events identified by URIs and entities | Event-event relations (temporal and causal relations), Event-entity relations (action, participant, time, and location), and general facts about entities | Instance level |
| Event evolutionary graph [27] | Events (i.e., abstract, generalized, and semantic complete verb phrases) | Temporal and causal relations between events | Schema level |
| Event-centric temporal KG [28] | events, entities, and relations | Event-event relations (subevent, previous, and next event relations), event-entity relations (topic, participant, time, and location), and entity-entity relations | Instance level |
| Event logic graph [29] | Abstract, generalized, and semantic complete event tuples (s, p, o) | Temporal, causal, conditional and hypernym-hyponym relations between events | Schema level |

logic graph are composite structures, which are difficult to handle. Moreover, these EKG-related concepts all consider specific and limited event relations and argument roles. Actually, there are many event relations. Besides, an event has its components, each of which consists of an argument and the argument role that the argument plays in the event [30].

2.2 Definitions

As introduced in Section 2.1, there are some EKG-related concepts with deficiencies. We follow this line but introduce richer contents, as presented in the following.

We think that EKG, centering on events, has two types of nodes, events and entities, and three types of directed edges, indicating event-event, event-entity, and entity-entity relations. As illustrated in Fig. 1, the first type of relations includes many kinds of relations between events, such as temporal, causal, conditional, thematic, etc. The second type of relations represents the arguments of events, i.e., the edges are the argument roles of the entities to the linked events. The third one describes the relations between entities, such as spouse, place of birth, country, etc. Formally,

Definition 1. EKG $G = \{(s, p, o) | \{s, p\} \in N, p \in E, N = N_{evt} \cup N_{ent}, E = E_{evt-evt} \cup E_{evt-ent} \cup E_{ent-ent}\}$ is a graph of events N_{evt} , entities N_{ent} , and their relations E , where $E_{evt-evt}$, $E_{evt-ent}$, and $E_{ent-ent}$ are the relations between events, between events and entities, and between entities, respectively.

In this way, events can easily be connected by common argument entities and vice versa. KG is thus a special case of EKG, with only entity nodes and entity-entity relations.

3 WHAT IS EKG: ONTOLOGY VIEW

From the ontology view, we look into the schema and related tasks. As presented in the bottom part of Fig. 1, EKG schema describes the basic concepts that form EKG, including the event types, argument roles, and event relations. The first two form event schema. As for the last one, the typical script [31] organizes a set of events via some event relations, which together describe the common scenario. Before introducing EKG schema induction, let us begin with event schema and script inductions in this section.

3.1 Event Schema Induction

Event schema can be manually designed, such as the typical ACE event schema [19], [30] and FrameNet frame [32]. Since manually designed event schema shows low coverage and difficulty in domain adaptation, researchers have paid attention to event schema induction. It automatically extracts event types and their argument roles from texts. Formally,

Definition 2. Event schema induction: Given a set of texts $\{T_0, T_1, \dots, T_l\}$, it identifies the event schema, including all event types $\{tp_0, \dots, tp_\tau\}$ and all argument roles $\{r^i_0, \dots, r^i_\rho\}$ for each event type tp_i ($0 \leq i \leq \tau$).

For example,

Example 1. Input: T_0 : Barack Obama previously served as a U.S. senator from 3 January 2005. Then, he was selected as the President of America on 20 January 2009.

T_1 : Before winning the presidential election, Obama was nominated for the President of America on 27 August 2008.

TABLE 2
Comparison of methods for event schema induction.

| Method category | Description | Strengths | Weaknesses |
|-----------------|--|---------------------------|----------------------------------|
| Supervised | Features combined with pattern matching or classifier | Simple and intuitive | Hard to apply to new event types |
| Semi-supervised | Apply heuristic rules or vector quantized variational autoencoder with a few annotated seeds | Low labor costs | Not so effective |
| Unsupervised | Apply rules, statistics, or graph analysis (combined with representation learning) | Efficient and widely used | More noises |

Output: tp_0 : Start-position, rl_0^0 : person, rl_1^0 : position, rl_2^0 : time; tp_1 : Nominate, rl_0^1 : nominee, rl_1^1 : position, rl_2^1 : time.

Existing methods for this task can be divided into supervised, semi-supervised, and unsupervised ones.

Supervised methods are applied in early studies. They learn from annotated data and then induce event schema from new texts [33], [34], [35], [36]. For example, methods in the third MUC evaluations used pattern matching (e.g., regular expressions), syntax-driven techniques which combine syntactic analyses with semantics and subsequent processing, or integrated syntax-driven techniques into pattern matching for event schema induction [34]. Chieu et al. [36] adopted semantic and discourse features, and built a classifier, e.g., maximum entropy, Support Vector Machine (SVM), naive Bayes, or decision tree, for each argument role.

Semi-supervised methods start with a few annotated seeds to induce event schema [37], [38], [39], [40]. For example, Patwardhan and Riloff [39] created a self-trained SVM to identify relevant sentences for the domain of interest and then extracted domain-relevant event schema via semantic affinity. The self-training begins with seed patterns and relevant and irrelevant documents. The following event schema extraction is based on heuristic rules upon syntactic analyses. The extracted results were ranked by semantic affinity based on frequency to keep the top results. Huang and Ji [40] discovered unseen event types by leveraging annotations available for a few seen types. They designed a vector quantized variational autoencoder to learn an embedding for each seen or unseen event type and optimized it using seen event types. A variational autoencoder was further introduced to enforce the reconstruction of each event trigger conditioned on its event type distribution.

Unsupervised methods remove the requirements of annotated data and are widely applied [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51]. For example, Chambers and Jurafsky [43] viewed event schema induction as discovering unrestricted relations. They used Pointwise Mutual Information (PMI) to measure the distances between events and clustered events according to the distances. Then, they induced the argument roles of events via the syntactic relations. Balasubramanian et al. [44] used co-occurrence statistics of (s, p, o) pairs to build a graph with these triples as nodes and edges were weighted by the symmetric conditional probabilities of the involved triple pairs. Triples therein were normalized using stemmed headwords and semantic types. They started with high-connectivity nodes as seeds. Then, they applied graph analysis to find closely related triples to the seeds and merged their argument roles to create event schema. Chambers [45] proposed the first

generative model for schema induction similar to LDA [52]. Nguyen et al. [46] further introduced entity disambiguation.

More recent studies introduced representation learning to induce event schema unsupervisedly [47], [48], [49], [50], [51]. For example, Yuan et al. [48] proposed a two-step framework. They first detected event types by clustering news articles. Then, they proposed a graph-based model that exploits entity co-occurrence to learn entity embeddings and clustered these embeddings into argument roles. Methods in the 2019 International Workshop on Semantic Evaluation applied pretrained language models, such as BERT [53], to get word embeddings [49]. They then clustered these embeddings with hand-crafted features and aligned them to the event types and argument roles of the existing event schema (e.g., FrameNet). Yamada et al. [50] thought previous studies focused too much on the superficial information of verbal event triggers and proposed to use masked word embeddings from BERT to get deep contextualized word embeddings. They then applied a two-step clustering method, which clusters instances of the same verb according to the embeddings and further clusters across verbs. Finally, each generated cluster was regarded as an induced schema.

In a word, as presented in Table 2, for supervised methods, they are hard to apply to new event types, which limits their usage. For semi-supervised and unsupervised methods, automatically derived event schema is noisy and difficult to be aligned. So far, these techniques are still not so applicable to building the event schema for an EKG.

3.2 Script Induction

A script can be seen as a stereotypical structure of event schema that expresses a specific scenario. Specifically, it organizes a set of events into a certain structure (usually according to their temporal relations). The “events” in scripts, called script events, are event schema rather than instances.

In early studies, scripts were manually designed [31], [54]. In more recent studies, researchers tried to extract script automatically from texts, i.e., script induction. Formally,

Definition 3. Script induction: Given a set of texts $\{T_0, T_1, \dots, T_l\}$, it identifies all the scripts $\{\Phi_0, \dots, \Phi_\epsilon\}$, and each script Φ_i ($0 \leq i \leq \epsilon$) is a coherent event schema sequence $\{tp_0^i(rl_0^{i,0}, \dots, rl_{n_i^0}^{i,0}), \dots, tp_\phi^i(rl_0^{i,\phi}, \dots, rl_{n_i^\phi}^{i,\phi})\}$, where n_j^i ($0 \leq j \leq \phi$) is the number of argument roles of event type tp_j^i , $rl_\theta^{i,j}$ is the θ -th argument role of tp_j^i .

Fig. 3 is a typical example of induced restaurant script, modeling the scenario of a customer eating food in a restaurant.

The main challenge is to learn the scenario-specific argument roles automatically. Chambers and Jurafsky [55] used

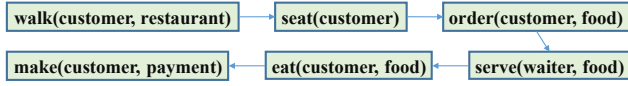


Fig. 3. A typical example of induced restaurant script.

the most frequent headword in an entity coreferential chain as the argument role of this entity. Regneri et al. [56] collected descriptions of script-specific event sequences from volunteers. They then computed a graph representation of the script's temporal structure. This graph makes statements about what phrases describe events of a scenario and in what order they occur. Cheung et al. [57] defined a joint distribution over the words in a document and their script assignments. Specifically, they combined a script Hidden Markov Model (HMM) [58] with an event HMM, where the first HMM models script transition and emits events, and the second one models event transition within a script and emits argument roles. Orr et al. [59] also learned scripts based on HMM. Differently, the states of the HMM are event types in scripts, and observations are sentences describing the event instances, which were obtained via a clustering algorithm. Then, they started with a fully enumerated HMM of the event sequences and merged states and deleted edges (state transitions) to improve the posterior probability of the structure and parameters given the data. Weber et al. [60] argued that correlation-based approaches are insufficient and induced script based on the causal effect between events, defined via interventions using a Bayesian network-based method. Weber et al. [61] and Ciosici et al. [62] further induced script via human-machine collaboration manner.

In general, among these script induction methods, the simple one proposed by Chambers and Jurafsky [55] is usually used in the studies of script application [63], [64], [65]. However, it is far from satisfactory. For example, suppose a coreferential chain contains "Jobs", "Steve Jobs", and "CEO Jobs". The above method will use "Jobs" instead of "CEO" as the argument role, which is too specific and lacks generalization. It is still a problem to learn high-quality scenario-specific argument roles effectively and efficiently.

Script can be seen as rules about events, forming the evolution patterns of events in a certain scenario. Its one basic application is to predict what will happen in the future via script event prediction. Specifically, the known real-world events are generalized into script events. Then, they are used to derive the subsequent script events, called script event prediction. Finally, the predicted subsequent script events can be instantiated to real-world events. We will describe more details of script event prediction in section 5.1.1.

3.3 EKG Schema Induction

There are few studies for EKG schema induction. Formally, **Definition 4. EKG schema induction:** Given a set of EKGs $\{G_0, \dots, G_\xi\}$, it identifies the schema $\mathbb{G} = \{\mathbb{E}, \mathbb{R}_{evt-evt}, \mathbb{R}_{ent-ent}\}$, where \mathbb{E} is the event schema, including event types and their argument roles constraining the arguments (i.e., event type-entity type relations), and $\mathbb{R}_{evt-evt}$ and $\mathbb{R}_{ent-ent}$ are the sets of relations between event types and between entity types, respectively.

Take Fig. 1 for a simple example, with the instance graph as input, EKG schema induction outputs the schema (see the bottom part of Fig. 1).

Since EKG-related studies have not been put forward for a long time, researches on EKG schema induction are much scarcer and newer. In 2020, Li et al. [66] first studied event graph schema induction, focusing on rich event components and event-event connections. A path language model was proposed to build an event graph schema repository, where two event types are connected through event-event paths involving entities that fill important argument roles. These entities were replaced by their types. However, this work only pays attention to the connections between events. In 2021, Li et al. [67] further focused on all three types of relations, i.e., event-event, event-entity, and entity-entity relations. They regarded schema as hidden knowledge to guide the generation of event graphs and learned via maximizing the probability of these instance graphs. However, for event-event relations, only temporal relations were considered.

Generally speaking, existing few studies on EKG schema induction consider limited relation types. Thus, there is a long way for EKG schema induction to induce overall schema for EKG.

4 WHAT IS EKG: INSTANCE VIEW

From the instance view, this section introduces how to construct an EKG, i.e., event acquisition and EKG-related representative graphs/systems.

4.1 Event Acquisition

Event acquisition is essential for EKG construction. As presented in Fig. 4, it mainly includes event extraction, event relation extraction, event coreference resolution, and event argument completion. The former two tasks are basic, and the latter two are important for constructing a better EKG. Due to the complex event structures, they are more challenging than entity extraction, entity relation extraction, and entity coreference resolution in KG construction. In this section, we review their development and future trends.

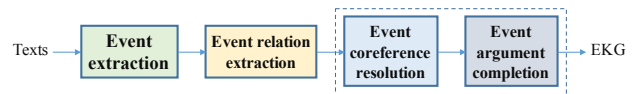


Fig. 4. Different steps in event acquisition.

4.1.1 Event Extraction

As the primary step in constructing EKG, event extraction is to extract structured event information from texts, including event triggers with types and arguments with roles. Thus, there are two subtasks, trigger detection and argument extraction. The former identifies the event trigger words and assigns them with predefined types or clustered classes, while the latter identifies the arguments and assigns them with argument roles of the triggered events. Formally,

Definition 5. Event extraction: Given a text $S = \{w_0, w_1, \dots, w_n\}$ of n words, it identifies all event triggers $\{trg_0, \dots, trg_m\}$ and predicts the event type tp_i for each event trigger $trg_i = \{w_{b_i^{trg}}, \dots, w_{d_i^{trg}}\}$, where $0 \leq i \leq m$, b_i^{trg} and d_i^{trg} are the beginning and end indexes of trg_i . It also identifies event arguments $\{arg_0^i, \dots, arg_k^i\}$ for trg_i and predicts the argument role rl_j^i for each event argument $arg_j^i = \{w_{b_{ij}^{arg}}, \dots, w_{d_{ij}^{arg}}\}$, where $0 \leq j \leq k$,

b_{ij}^{arg} and d_{ij}^{arg} are the beginning and end indexes of arg_j^i .

For example,

Example 2. Input: S : Barack Obama previously served as a U.S. senator from 3 January 2005.

Output:

| | |
|----------------------------|--------------------------|
| trg_0 : served as | tp_0 : Start-position; |
| arg_0^0 : Barack Obama | rl_0^0 : person; |
| arg_1^0 : U.S. senator | rl_1^0 : position; |
| arg_2^0 : 3 January 2005 | rl_2^0 : time. |

Event extraction is divided into schema-based and schema-free ones concerning if there is a predefined schema. As illustrated in the upper part of Fig. 5, existing schema-based methods pass texts to feature learners to obtain local (and global) features. Upon them, classifiers of triggers and arguments output the probability distributions on the predefined schema and get the answers based on the peaks. As for schema-free event extraction formalization (the bottom part of Fig. 5), texts are passed to the discriminators to get raw triggers and arguments, which are clustered into groups to induce the event schema and get the answers. Simple unsupervised event schema induction methods (see Section 3.1) are usually used therein. Specifically, considering the input scale, schema-based event extraction is further grouped into sentence- and document-level ones, and schema-free event extraction is also called open-domain one. Compared to document-level and open-domain event extractions, sentence-level one is more extensively studied.

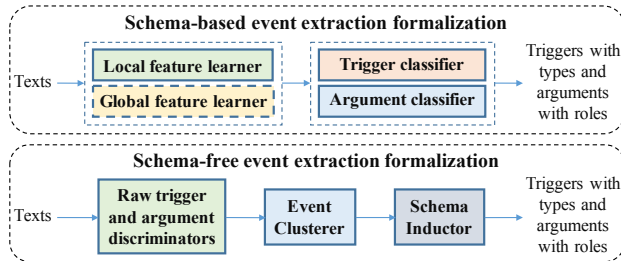


Fig. 5. Existing formalizations of event extraction.

Sentence-level event extraction extracts event triggers and arguments in a sentence. Early approaches designed elaborate features and applied statistical learning methods [68], [69], [70]. Recently, with the construction of large-scale datasets (e.g., ACE [30], TAC-KBP [71], and RAMS [72]) and the development of deep learning, researchers have adopted neural networks to extract features automatically. According to the feature scopes, they are divided into three categories:

Explore intra-subtasks features. Most methods employ a Convolutional Neural Network (CNN) or Recurrent Neural Network (RNN) to extract the intra-subtasks features and follow a pipeline framework performing each event extraction subtask sequentially. Chen et al. [73] proposed dynamic multi-pooling CNN to extract the word and various sentence-level features for candidate words. However, as a CNN-based method, this work cannot properly handle the sequential relations and long-range dependencies among the words in a sentence. Thus, Chen et al. [74] adopted bidirectional dynamic multi-pooling Long Short-Term Memory Network (LSTM). Besides, they designed a tensor layer to explore interactions between candidate arguments. With the development of the pretrained language models like BERT

and ELMo [75], recent researchers have introduced them into event extraction. For example, Yang et al. [76] added a multi-classifier on the BERT for trigger detection and multiple sets of binary classifiers on the BERT for argument extraction. Deng et al. [77] enriched event schema with event-event relations, such as temporal, causal, and hierarchical relations, to improve trigger detection based on BERT. Recently, Du and Cardie [78] and Liu et al. [79] proposed machine reading comprehension frameworks upon BERT, which adopt question answering to extract events.

Explore inter-subtasks features. These methods explore the inter-subtasks features of trigger detection and argument extraction, and usually follow a joint framework, easing error propagation of the above pipeline framework. Nguyen et al. [80] proposed a joint model based on bidirectional gated recurrent units and Fully Connected Network (FCN). They introduced memory vectors to store the dependencies among trigger subtypes and argument roles. To further make use of syntactic features, Sha et al. [81] added dependency bridges to connect syntactically related words in RNN and built a tensor layer on each pair of candidate arguments to capture intensive argument-level interaction. To capture the long-range dependencies more efficiently, Liu et al. [82] used an attention-based Graph Convolution Network (GCN) to aggregate the word information through the paths of the syntactic tree of the sentence and capture more interactions among the candidate triggers and arguments.

Explore inter-IE features. To better model the semantic information of event elements, recent researchers have explored more global features via introducing other related Information Extraction (IE) tasks. With a multi-task learning framework, these IE tasks benefit from each other. Nguyen and Nguyen [83] employed a bidirectional RNN to learn word embeddings, over which entity and event extractions were conducted via the classifiers based on FCN and softmax. Wadden et al. [84] further handled entity extraction, entity relation extraction, event extraction, and coreference resolution. After encoding sentences via BERT, they enumerated text spans and constructed a span graph. Span embeddings were updated by integrating embeddings from their neighbors and passed to the FCN-based classifiers of all the tasks. However, these studies handle the tasks separately. To better explore the inter-IE features, Lin et al. [85] extracted entities, entity relations, and events simultaneously. They computed local scores for all the candidate triggers, entities, and their pairwise links via the BERT embeddings and FCN-based classifiers. Then, they searched for the globally optimal results with a beam search-based decoder by incorporating inter-dependencies among the candidate triggers and entities.

Document-level event extraction. Sentence-level event extraction assumes that event trigger and its arguments are in the same sentence. However, they usually scatter across multiple sentences in a document in real-world scenarios. Thus, document-level event extraction is practical. It is also more challenging, since arguments may exist in different sentences and a document usually contains multiple events.

Early approaches used handcrafted features to model events and involved entities. Then, they extracted events in the document via statistical learning methods [86], [87], [88]. These features and annotated data are usually expensive

TABLE 3
Comparison of methods for event extraction.

| Method category | Description | Strengths | Weaknesses |
|---|--|---|----------------------------------|
| Sentence-level event extraction: Explore intra-subtasks features | Follow a pipeline framework performing each subtask sequentially | Intuitive and with low model complexity | Error propagation |
| Sentence-level event extraction: Explore inter-subtasks features | Follow a joint framework performing each subtask simultaneously | Capture the dependencies among event elements | High model complexity |
| Sentence-level event extraction: Explore inter-IE features | Adopt a multi-task learning framework introducing other related IE tasks | Benefit from more global features and is more effective | High model complexity |
| Document-level event extraction | Extract events in multiple sentences via the local and global features | Fit real-world scenarios | Less studied and far from solved |
| Open-domain event extraction | Extract simple event elements through clustering without a predefined schema | No annotation data | Less effective |

to obtain. To address this problem, Yang et al. [89] used a distantly supervised method to label events in the document automatically. With the annotated data, they trained a sentence-level event extraction model with a Bidirectional LSTM (BiLSTM) and a Conditional Random Field (CRF) layers. Based on the extracted events, they padded the missing arguments. This framework tackles argument scattering in a pipeline way. To handle document-level event extraction end to end, Zheng et al. [90] transformed events into an entity-based directed acyclic graph. Then, they transformed event extraction into several sequential path-expanding subtasks. Xu et al. [91] also did this transformation but explored more interactions among sentences and events via applying Graph Neural Network (GNN) on the graph of sentences and entities and storing the extracted events into a global memory to facilitate the current extraction. Differently, Lou et al. [92] proposed a multi-layer bidirectional network to capture document-level information and event inter-dependency simultaneously for trigger detection.

Open-domain event extraction. Unlike sentence- and document-level event extractions, open-domain one has no predefined schema. It extracts events mainly from long texts like newswires and short ones like social media streams.

For long texts, Rusu et al. [93] treated verbs as event triggers and analyzed the dependency paths between verbs and other syntactic elements (entities, time expressions, subjects, and objects) to identify arguments. However, syntactic relations are too simple to describe complex events. To extract events and induce schema simultaneously, Huang et al. [47] identified all the nouns and verbs that can be matched to the existing schema like FrameNet as candidate event triggers and identified their candidate arguments via manually-selected semantic relations. Then, they were clustered based on their embeddings from a tensor-based model and named to obtain the schema and extracted results via the mappings to the existing schema. Differently, Liu et al. [94] learned the joint probability distribution of a news cluster with headwords, contextual features, and latent event type. Then, they adopted the learned distributions to cluster news and applied a series of rules to get the final results.

Unlike long texts that need inducing a complex schema, the main event components in social media streams these short texts are entities, time, location, and keywords. Abdelhaq et al. [95] detected events important in a small area from a tweet stream. They extracted event keywords based on frequency. Then, they calculated the spatial density

distributions over the usage ratios of the keywords at a particular location. The ones with small entropy (occur at a few locations) were clustered by their distributions to get the events. To extract more information, Wang et al. [96] fused Twitter and related Web pages to extract events, their times, locations, and titles by tweet- and page-based CRFs, respectively. However, multiple mentions may refer to the same entity and would be wrongly assigned to different events by the above methods. Thus, Zhou et al. [97] proposed a non-parametric Bayesian mixture model for event extraction from Twitter, which applies word embeddings to deal with this issue. Xu et al. [98] further used a BiLSTM, a control gate, and a CRF layers to extract events from Twitter.

In a word, although event extraction has been studied for a long time (see Table 3), the performance is unsatisfactory, especially for argument extraction. Thus, it may introduce noises into EKG construction. Moreover, partly due to the low performance of argument extraction, some researches on EKG applications like script event and temporal KG predictions (see Sections 5.1.1 and 5.1.2) only consider the simpler fixed argument roles, subject, object, and indirect object or time, instead of well-defined complex schema. Thus, improving the performance of this primary event extraction task is of great importance.

4.1.2 Event Relation Extraction

Besides event extraction, event relation extraction is fundamental to EKG construction. It extracts relations between events from texts and links events to get an EKG. Formally,

Definition 6. Event relation extraction: Given a text $S = \{w_0, w_1, \dots, w_n\}$ of n words, where a pair of events are specified as (e_1, e_2) , it identifies their relation $r(e_1, e_2)$.

Currently, events are simplified as verbs. For example,

Example 3. Input: S : Barack Obama previously $\langle e_1 \rangle$ served as $\langle /e_1 \rangle$ a U.S. senator from 3 January 2005. Then, he was $\langle e_2 \rangle$ selected as $\langle /e_2 \rangle$ the President of America on 20 January 2009.

Output: $r(e_1, e_2)$: before.

The main event relations are event temporal and causal relations. For short, we use the terms temporal and causal relations, respectively. The former describes the temporal order between events. The latter describes the causality between events and is a subset of temporal relation. These two types of relation extractions share similar research lines and are usually formalized as text classification given the event

pairs and contexts. As presented in Fig. 6, extracted event pairs and their contexts are passed to the feature learner to capture helpful information. Based on these features, the relation classifier outputs the relation labels. Some methods also introduce external knowledge. For temporal relation extraction, some methods consider the important global consistency problem. For example, if the classifier gets the results: *A before B*, *B before C*, and *C before A*, then there is a conflict. Actually, the first two imply *A before C*.

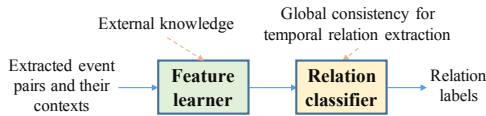


Fig. 6. Existing formalization of event relation extraction.

Early methods relied on the manually designed syntactic and semantic features [99], [100], [101], [102], [103], [103], [104]. They employed machine learning models, such as naive Bayes, maximum entropy, or SVM as the classifier to recognize relations. For example, Chambers et al. [101] applied SVM as the classifier for temporal relation extraction. Features therein include temporal attributes of events, such as tense, event type, modality, polarity, and other linguistic features. Rink et al. [103] used graph patterns as features to train an SVM classifier. Specifically, they built a graph representation for a sentence that encodes lexical, syntactic, and semantic information. They automatically extracted graph patterns from such graph representations, sorted by the relevance in determining the causality between events. Zhao et al. [104] proposed a restricted hidden naive Bayes model to handle feature interactions for causal relation extraction. Besides contextual, syntactic, and position features, they utilized a new feature of causal connectives. It is obtained from the similarities of the syntactic dependency structures of sentences expressing causality.

The above methods for temporal relation extraction only focus on pairwise decisions. Thus, many methods further considered global consistency [21], [23], [105], [106], [107], [108], [109]. Some applied Integer Linear Programming (ILP) on top of the above local classifiers [21], [23], [105]. Chambers et al. [107] proposed a cascade architecture with a sequence of classifiers ordered by their precision. The classifiers were run in order starting with the most precise one. Global consistency was enforced by inferring all the transitive relations from the results of earlier classifiers before passing them to the next classifier. Mirza and Tonelli [108] further applied similar methods as temporal and causal relation extractors. Different from these pipeline manners, some methods considered global consistency in the learning stage [106], [109]. Yoshikawa et al. [106] proposed a Markov logic model and captured global consistency via the addition of weighted first-order logic formulae. Ning et al. [109] trained the local classifier with feedback by performing global inference in each round of the learning.

Besides the above intra-sentence causal relation extraction, Gao et al. [110] designed lexical, potential causal, and syntactic features for document-level one. These global and fine-grained aspects of causality were learned via ILP.

More recent methods utilized neural networks to learn useful features for extraction automatically [111], [112], [113], [114], [115], [116]. They applied CNN or LSTM to en-

code event sentences, followed by a relation classifier based on FCN and softmax. To further make globally consistent decisions for temporal relation extraction, Han et al. [114] adopted an SVM-based algorithm incorporating transitivity constraints, and Ning et al. [115] employed ILP. Differently, Cheng and Miyao [117] adopted BiLSTM along dependency paths of event sentences for temporal relation extraction.

To further make use of external knowledge, Ning et al. [115] applied a Siamese network to a temporal common sense knowledge base, whose output was concatenated with the LSTM output of event sentences, for temporal relation extraction. Li and Mao [116] proposed a knowledge-oriented CNN for causal relation extraction, where the filters were generated from lexical knowledge bases to represent causal keywords and cue phrases. They also combined a conventional CNN to learn other features of causal relation.

To further introduce other related tasks, Han et al. [118] extracted event and temporal relation jointly to avoid error propagation in the pipeline manners, which extract them sequentially. Specifically, after encoding sentences via BiLSTM, they computed the probability of being an event and the softmax distribution over all the temporal relation labels. Global consistency was considered via the last SVM-based layer incorporating constraints of transitivity, symmetry, etc. Wang et al. [119] proposed a similar method for temporal and subevent relation extractions but replaced the SVM-based layer with a differentiable constrained learning layer. The constraints were converted into differentiable functions. Different from these hard constraints, Han et al. [120] improved similar networks of event and temporal relation extractions by introducing corpus statistics as soft constraints.

Notably, since pretrained language models perform well on many NLP tasks, researchers have introduced them, such as BERT, into event relation extraction. Many neural network-based researches simply used them as pretrained word embeddings [114], [115], [118], [119], [120]. Differently, some recent methods adopted BERT to encode event sentences [121], [122]. For example, Liu et al. [121] proposed knowledge enhanced event causal relation extraction with masking generalizations. The model consists of the knowledge-aware reasoner, masking reasoner, and attentive sentinel trading off between them. The first module uses BERT to model sentences where events are replaced by definitions from external knowledge, learning expressive event embeddings. The second one also applied BERT on sentences, but event mentions are replaced by [MASK] symbol, mining event-agnostic and context-specific patterns.

Recently, some BERT-based studies tackled data lacking in causal relation extraction via introducing external knowledge [123], [124], [125], [126]. Zuo et al. [123] proposed a knowledge enhanced distant data augmentation framework. They extracted causal event pairs based on lexical knowledge and used the results to label sentences distantly, which were further refined with causal commonsense knowledge. Then, they employed relabeling and annealing strategies to make use of distantly labeled sentences to train the causal relation extraction model based on the BERT encoder and FCN-based classifier. Zuo et al. [124] further proposed a knowledge-guided and learnable data augmentation framework. They regarded causal relation extraction and sentence generation as dual tasks and modeled the

TABLE 4
Comparison of methods for event relation extraction.

| Method category | Description | Strengths | Weaknesses |
|---------------------------------|--|---|--|
| Feature-based | Use manually designed syntactic and semantic features | High precision | Difficult to transfer to other domains |
| Neural network-based | Automatically learn features via neural networks | Easy to transfer | Require large data |
| Pretrained language model-based | Adopt pretrained language models to encode event sentences | Easy to transfer and utilize external knowledge from large corpus | Heavy computation |

mutual relations via dual learning. The generation was initialized with causal event pairs from external knowledge, ensuring the causality of generated sentences. Both causal relation extraction and sentence generation therein were based on BERT. Differently, Zuo et al. [125] learned context-specific causal patterns from external causal statements. Then, they adopted a contrastive transfer strategy to incorporate the learned patterns into the target causal relation extraction model based on BERT. Cao et al. [126] concatenated contextualized embeddings from the BERT of event sentences, GCN embeddings of one-hop neighbors of events from external knowledge, and densely connected GCN embedding of the shortest multi-hop path between the event pair from external knowledge. They then passed the concatenated embeddings to the FCN-based classifier.

The above BERT-based causal relation extraction methods are limited to the intra-sentence setting. For document-level setting, Phu and Nguyen [127] proposed a graph-based model. They applied BERT to encode words in the document, which were used to generate an interaction graph for the document considering the discourse, syntax, and semantic information. This graph was then consumed by GCN to learn document context-augmented embeddings for causality extraction based on FCN and softmax.

In general, existing event relation extraction methods (see Table 4) cannot fully satisfy the requirements of EKG construction. For example, they usually only focus on verbs as events and do not consider nouns. Actually, event triggers can be verbs or nouns. Another limitation is that they ignore arguments. Future researches on event relation extraction should pay attention to these fundamental problems.

4.1.3 Event Coreference Resolution

There are usually many texts describing the same events. It is necessary to group the events referring to the same real-world event into the same cluster after event extraction. This task is called event coreference resolution. Formally,

Definition 7. Event coreference resolution: Given a set of texts $\{T_0, T_1, \dots, T_l\}$, each text T_i ($0 \leq i \leq l$) contains some events $\{e_i^0, e_i^1, \dots, e_i^\alpha\}$, it divides these events into clusters $\{C_0, C_1, \dots, C_\beta\}$, where each cluster consists of events that refer to the same real-world event.

For example,

Example 4. Input: T_0 : Barack Obama previously served as a U.S. senator from 3 January 2005. Then, he was selected as the President of America on 20 January 2009.

T_1 : Before winning the presidential election, Obama represented Illinois in the U.S. Senate from 2005 to 2008.

Output: Two event clusters: $C_0 = \{\text{served as, represented}\}$ and $C_1 = \{\text{selected as, winning}\}$.

Event coreference resolution is divided into within- and cross-document settings according to whether events are from the same document or different ones. The latter is more intricate, since it is difficult to deal with event contexts from different documents. For example, semantically similar event contexts from different documents may describe different events. As illustrated in Fig. 7, existing methods pass the results from event extraction and their contexts to the feature learner and coreference scorer to get the coreference results between events. Then, a cluster decoder is applied to merge the local results to get the global ones, where some rules or clustering algorithms are adopted. Some methods additionally introduce external knowledge to improve feature learning. Specifically, existing methods can be divided into unsupervised, semi-supervised, and supervised ones.



Fig. 7. Existing formalization of event coreference resolution.

Unsupervised methods construct feature template-based event representations and adopt pattern matching or unsupervised probabilistic models to identify coreference relations [128], [129], [130], [131], [132], [133], [134], [135].

Early researches adopted rule-based approaches to deal with event coreference resolution [128], [129]. They applied the consistency of event triggers and arguments to determine whether two events are coreferential. Subsequently, some studies [130], [131], [135] used lexical features of event triggers, argument-related features, semantic features, and other handcrafted ones to construct event representations. Then, maximum entropy, non-parametric Bayesian estimation, cosine similarity, and other methods were adopted to determine the coreference of event pairs. Moreover, Chen and Ng [133] proposed an unsupervised probabilistic model for event coreference resolution and further introduced anaphoricity determination. The lexical and semantic features of triggers and arguments were also applied, and the expectation-maximization algorithm was used to estimate the model's parameters. These unsupervised methods can usually handle both within- and cross-document settings.

Semi-supervised methods pay attention to the scarcity of existing annotated corpus [136], [137], [138], [139]. They use a small amount of labeled data and a large amount of unlabeled data to conduct event coreference resolution.

For example, Sachan et al. [137] proposed an active learning-based event coreference resolution method. Some heuristic sample selection strategies, such as the maximum uncertainty, maximum expected judgment error, and explo-

TABLE 5
Comparison of methods for event coreference resolution.

| Method category | Description | Strengths | Weaknesses |
|-----------------------------------|--|--|--|
| Unsupervised | Construct feature template-based event representations and perform pattern matching | Intuitive and efficient | Limited scalability |
| Semi-supervised | Make use of unlabeled data or external resources to enhance the coreference scorer | Low labor costs | Not so effective and external resources may introduce noises |
| Supervised (event-pair models) | Process event pairs and adopt binary classifier as coreference scorer | Effective and handle cross-document setting well | High computational complexity |
| Supervised (event-ranking models) | Process the antecedent events simultaneously and find the first coreferential antecedent | Use more context | Less suitable for cross-document setting |

ration and exploitation, were used to choose event pairs for manual annotating. Peng et al. [138] conducted trigger detection and event coreference resolution in a unified framework. For the former, the similarity between candidate trigger and event type was adopted, where event type embedding is the embedding average of its event examples. For the latter, the similarity between event embeddings was applied. Event embedding was obtained via concatenating the embeddings of its elements trained from external texts. Their similarity thresholds were tuned by the given few labeled samples. These semi-supervised methods can usually be applied to both within- and cross-document settings.

Supervised methods. With the construction of datasets such as MUC [18], ACE [30], and ECB/ECB+ [132], [140], and the development of the TAC KBP Event Nugget Detection evaluation task [71], researchers have developed many supervised methods for event coreference resolution. They are divided into event-pair and event-ranking models depending on the sample form of the coreference scorer.

Event-pair models are common and influential. They process event pairs and adopt binary classifier as coreference scorer to assign each event pair a probability of being coreferential [141], [142], [143], [144], [145], [146], [147].

For example, Krause et al. [143] adopted CNN as the feature learner to process event sentences, whose outputs were concatenated with the embeddings of event triggers and their left and right neighbors to get event embeddings. Two event embeddings were concatenated and augmented with their features, then passed to an FCN, followed by a logistic regression classifier as the coreference scorer. This method only focuses on within-document event pairs. To handle both within- and cross-document settings, Choubey and Huang [144] separately trained two neural network-based feature learners and FCN-based coreference scorers. Then, they alternated within- and cross-document cluster merging to model the second-order inter-dependencies across events. To further capture the semantic interactions between event contexts, Zeng et al. [147] proposed an interaction-based within- and cross-document coreference model. Specifically, two sentences of the event pair were concatenated and fed to the feature learner based on BERT. Meanwhile, the internal structure of the events was injected via semantic role label embeddings. FCN and softmax were used as the coreference scorer. Lee et al. [141] and Barhom et al. [146] further introduced entity coreference resolution to enhance the performance by interacting the two tasks. Lexical resource or pretrained word embeddings were used therein.

Event-ranking models process all events mentioned before

the given event, i.e., the antecedents, simultaneously. They are trained to rank the first coreferential antecedent of each given event first [148], [149], [150], [151], [152], [153], [154].

For example, Lu and Ng [149] selected the coreferential antecedent for each event in a document collectively. They defined the antecedent vector, where the i -th element is the coreferential antecedent index of the i -th event in the document. A log-linear model was employed to assign the highest score to this antecedent vector. Since coreference may be long-distance, document-level information is helpful. Thus, Tran et al. [153] constructed a structure graph of the events, entities, and words for each document, and applied GCN on it. Event embeddings from GCN were fed to the FCN-based coreference scorer. To make use of the cross-task interactions for better performance, many methods further incorporated trigger detection, entity coreference resolution, anaphoricity determination, realis detection, or argument extraction into a joint learning framework [148], [150], [151], [152], [154].

Notably, since the event-ranking models usually need more context, they are more suitable for within-document setting and may introduce noises for cross-document one.

Generally speaking, existing researches on event coreference resolution (see Table 5) still have some deficiencies. For example, most methods specify that all events have fixed arguments. However, arguments differ event by event. In addition, some methods only consider within-document setting and cannot handle cross-document one, while both are important for better EKG construction. Thus, future researches should develop practical methods.

4.1.4 Event Argument Completion

Since information in original texts is incomplete, and there are some missing in event extraction, the extracted events usually miss some elements. Event argument completion thus aims to complete existing events, generally formalized as filling in a missing argument or argument role. Formally,

Definition 8. Event argument completion: Given EKG G , it is to fill in event information for each incomplete event e , including filling in the argument arg^* for a target argument role tgt_{rl} , $arg^* = \arg \max_{arg} \Pr(arg|e, tgt_{rl}, G)$ and filling in the argument role rl^* for a target argument tgt_{arg} , $rl^* = \arg \max_{rl} \Pr(rl|e, tgt_{arg}, G)$.

For example,

Example 5. Input: {event type: Start-position, person: Barack Obama, position: U.S. senator}, target argument role "time", and other existing events.

Output: 3 January 2005.

TABLE 6
Comparison of methods for event argument completion.

| Method category | Description | Strengths | Weaknesses |
|-------------------------|---|--------------------------|-----------------------|
| Learn from thematic fit | Combine the known elements and further the target one based on neural networks to predict the missing one | Simple | Not so effective |
| Based on graph sequence | Design the score learner upon the time-specific embeddings of event elements | Good at interpretability | Need time to be known |
| From n-ary facts | Evaluate the composability of event elements | Good at interpretability | High model complexity |

Existing methods further formalize this task as a prediction or classification task. As shown in Fig. 8, the former passes known elements to the feature learner and filler predictor to get the predicted element. It is compared with all the arguments or argument roles to get the overall distribution with the peak as the answer. The latter also inputs the candidate filler and learns the score of the candidate event. The candidate filler of the maximum score is the answer.

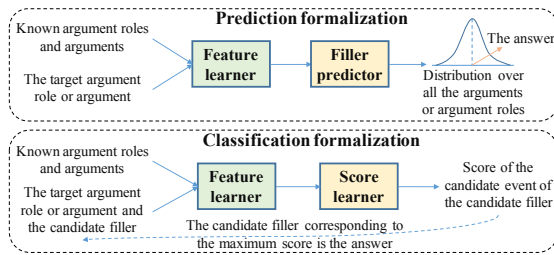


Fig. 8. Existing formalizations of event argument completion.

Some early researches learned from thematic fit for sentence comprehension, which determines the goodness of fit between the entities and the agent and patient roles of the verb [155]. For example, Tilk et al. [156] combined the embeddings of argument role-argument pairs based on neural network. They added these combined embeddings of known pairs to further combine with the target argument role and then predicted the missing argument. Hong et al. [157] replaced the sum of the pair embeddings with their weighted sum and additionally predicted the missing argument role for the target argument.

EKG with time can be reorganized as a sequence of graphs by time. Event argument completion is based on the score learner upon the time-specific embeddings of event elements in the graph sequence. García-Durán et al. [158] and Leblay and Chekol [159] integrated the time into event type embeddings by concatenating their embeddings. Then, the score learner was designed on these time-aware embeddings of event types and the embeddings of other arguments as their combination like TransE [160]. Dasgupta et al. [161] assigned each timestamp with a hyperplane. Events valid at a certain time were projected onto the corresponding hyperplane, where the translational scores of arguments via event type [160] were used for completion. Xu et al. [162] further maintained the smoothness between adjacent hyperplanes by minimizing their Euclidean distances. Differently, Lacroix et al. [163] represented an EKG as a tensor of event types, time, and other arguments. They then performed tensor decomposition and used the reconstructed tensor to conduct completion. Besides events with exact occurring time, TeRo [164] additionally handled events with beginning and end time, and accordingly represented each event type as a pair of dual complex embeddings. They

mapped other arguments to these two event type embeddings and combined the translation-based scores [160].

Events are typical n-ary facts. Completion methods for n-ary facts can also be used to event argument completion. Early studies applied translation-based methods. Wen et al. [165] defined the score learner for each candidate event as the weighted sum of the projection results from its arguments to the hyperplane of its event type, where the weights are the real numbers projected from its argument roles. Zhang et al. [166] additionally introduced the likelihood that two arguments co-participate in a common event via FCN. Liu et al. [167] further considered the relatedness among argument roles and the compatibility between each argument role and all the involved arguments. Subsequently, tensor-based methods were adopted. Liu et al. [168] represented events as a tensor and reconstructed it to make completion. Di et al. [169] further solved data sparsity via partially sharing embeddings and over-parameterization by sparsifying the tensor. Recently, neural network-based methods have sprung up. They used CNN [170], [171], [172], [173], FCN [170], [173], [174], GNN [175], [176], and Transformer [175], [177] to learn features and obtain scores of candidate events or fillers. For example, Guan et al. [170] adopted CNN to get the embeddings of argument role-argument pairs. The relatedness of these pairs via FCN was used to estimate the scores of candidate events. Galkin et al. [175] organized events into a graph and used GCN as the feature learner to get the embeddings of event elements. The learned embeddings of the known elements were passed to a Transformer-based filler predictor to get answer distribution.

On the whole, as summarized in Table 6, existing studies focus on event itself, while ignoring event relations. Introducing event relations may be helpful. It is an interesting direction in the future. Moreover, researchers simplify event argument completion as inferring a missing argument or argument role. However, the argument role and its argument are usually both missing. Thus, future studies should pay attention to more realistic formalizations and methods.

4.2 EKG-related Representative Graphs/Systems

With the development of event acquisition, there are some EKG-related representative graphs/systems targeting a specific or general domain.

Domain-specific graphs/systems. In 2016, Rospocher et al. [26] constructed four event-centric KGs, i.e., WikiNews in English, FIFA WorldCup in English, Cars in English, and Airbus Corpus in English, Dutch, and Spanish, from different news. They have over 624 thousand, 9.3 million, 25 million, and 2.5 thousand events, respectively. Some specific event-event and event-entity relations are considered as

TABLE 7
Summarization of EKG-related representative graphs/systems.

| Graph/system | Domain | Description | Scale |
|---|--|--|--|
| WikiNews, FIFA WorldCup, Cars, and Airbus Corpus [26] | General news, football, automotive industry, and airbus A380 | Focus on event temporal and causal relations, and events have actions, participants, time, and locations | Over 624 thousand, 9.3 million, 25 million, and 2.5 thousand events, respectively |
| Chinese travel domain event evolutionary graph [27] | Tourism | Temporal relations between events (verb phrases) | - |
| Chinese financial domain event logic graph [29] | Finance | Temporal, causal, conditional and hypernym-hyponym relations between events $((s, p, o)$ tuples) | Over 1.5 million events and their 1.8 million relations |
| Event-centric Hainan tourism KG [4] | Tourism | Temporal and spatial dynamics of tourists' trips | Over 7 thousand journeys, about 87 thousand events, about 141 thousand entities, and near 228 thousand edges of event arguments and about 80 thousand event temporal relations |
| EventPlus [178] | Multiple domains | System: Extract event triggers, arguments, duration, and temporal relations | - |
| EventKG [28], EventKG+Click [179], and OEKG [180] | General domain | Focus on subevent, previous event, and next event relations, and events have topics, participants, time, and locations | Over 690 thousand events, -, and over 436 million triples |
| CogCompTime [181] | General domain | System: Extract time expression, event triggers, and trigger temporal relations | - |
| ASER [182] | General domain | Temporal, contingency, comparison, expansion, and co-occurrence between eventualities (dependency graphs) | Over 194 million eventualities and 64 million relations |

introduced in Section 2.1. In 2017, Li et al. [27] constructed a Chinese travel domain event evolutionary graph from a large-scale unstructured Web corpus. Its nodes are events (simplified as verb phrases), and edges are sequential relations between events with transition probability. In 2019, Ding et al. [29] constructed a Chinese financial domain event logic graph from plenty of news¹. It focuses on event causal relations and has over 1.5 million event nodes (i.e., (s, p, o) tuples) and their 1.8 million directed edges. In 2020, Wu et al. [4] constructed an event-centric tourism KG based on touristic data in Hainan to model the temporal and spatial dynamics of tourists' trips. Its nodes are over 7 thousand journeys, about 87 thousand events, and about 141 thousand entities, while its near 228 thousand edges are event arguments and about 80 thousand event temporal relations. Each event contains three components, activity, time, and place, and is connected to its journey via the relation "contain". In 2021, Ma et al. [178] presented the event pipeline system EventPlus², with comprehensive event understanding capabilities to extract event triggers, arguments, duration, and temporal relations. It was designed with multi-domain support by multi-domain training. However, in its temporal relation graph, nodes are event triggers, and edges are their temporal relations. The event arguments are separate.

General-domain graphs/systems. In 2018, Gottschalk and Demidova [28] constructed a multilingual event-centric temporal KG EventKG from structured and semi-structured data, and some event-event, event-entity, and entity-entity relations were considered (see Section 2.1). It has over 690 thousand events. It was further extended to EventKG+Click [179] by introducing user interactions

with events, entities, and their relations, derived from the Wikipedia clickstream. Besides, based on EventKG, Gottschalk et al. [180] built OEKG (Open EKG) with over 436 million triples by further integrating event-related data sets from multiple application domains, such as question-answering, entity recommendation, and named entity recognition. Also in 2018, Ning et al. [181] proposed the temporal understanding system CogCompTime to extract time expression, event triggers, and temporal relations between event triggers. Argument information was not considered therein. In 2020, Zhang et al. [182] developed ASER (Activities, States, Events, and their Relations), an English eventuality KG extracted from reviews, news, forums, social media, movie subtitles, and e-books. In ASER, each node is an eventuality, which is a dependency graph, and each edge is a relation between eventualities. In the dependency graph, nodes are the words in the sentence, and edges are their dependency relations. Five categories of eventuality relations were considered, i.e., temporal, contingency, comparison, expansion, and co-occurrence. The full version of ASER has over 194 million eventualities and 64 million relations.

Thus, there are some EKG-related representative graphs or systems, summarized in Table 7. However, they all consider specific and limited event-event relations or argument roles. Actually, there are various event-event relations in real-world scenarios. Moreover, different events usually do not share argument roles. There is a need to develop practical EKGs to facilitate downstream applications in the future.

5 WHAT IS EKG: APPLICATION VIEW

EKG has substantial application value, with the introduction of events and event-event relations. This section introduces its basic and deep applications.

1. See <http://elg.8wss.com/main>
2. See <https://kairos-event.isi.edu/>

5.1 Basic Applications

The main basic applications are predictions on EKG, which predict future events based on the current EKG. There are two ways to deal with these predictions. The first one generalizes event instances to script events and predicts the subsequent script events at the script level, called script event prediction [183]. The predicted script events can then be instantiated to real-world events. The second one predicts future events at the instance level directly. Specifically, existing methods simplify EKG to temporal KG, formalized as a sequence of KGs of (s, p, o) with timestamps. Then, future prediction is to predict events for future timestamps, called temporal KG prediction [184], [185], [186], [187].

5.1.1 Script Event Prediction

Script event prediction is to predict the subsequent script events given the historical scripts. In the following, script event is denoted as event for convenience. Formally,

Definition 9. Script event prediction: Given the historical events $\mathcal{H} = \{e_0, e_1, \dots, e_x\}$, script structure $\mathcal{S} = \{r_0(e_{h_0}, e_{t_0}), r_1(e_{h_1}, e_{t_1}), \dots, r_y(e_{h_y}, e_{t_y})\}$, and candidate events $\mathcal{C} = \{e_{c_0}, e_{c_1}, \dots, e_{c_z}\}$, it predicts the most possible subsequent event $e^* = \arg \max_{e \in \mathcal{C}} \Pr(e|\mathcal{H}, \mathcal{S})$.

A typical example is presented in Fig. 9, where $Context(e_i)$ is the script structure for the subsequent event e_i to be predicted. The relations are all temporal one and omitted.

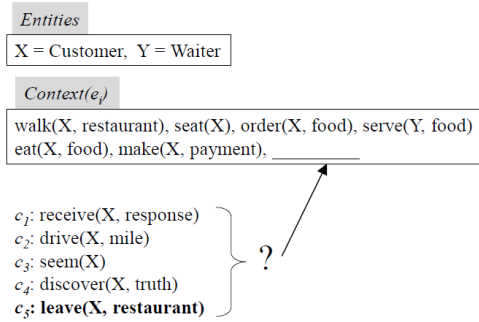


Fig. 9. A typical example of script event prediction [64].

Script event prediction is formalized as script coherence evaluation, and the event corresponding to the max coherence score is chosen as the answer. Chambers and Jurafsky [183] represented events as $(subject, predicate)$ or $(predicate, object)$ pairs. Given an event, they obtained its coherence score by aggregating its coherence scores with all the events from the script based on PMI. The narrative cloze test was adopted for evaluation. In this test, one event in the script is masked, and the model is asked to predict it. Following studies improve this model in three aspects:

Event representation. The pair representation in [183] loses the co-occurrence information of subjects and objects. Thus, Balasubramanian et al. [44] applied $(subject, predicate, object)$ triples. Pichotta and Mooney [188] and Granroth-Wilding and Clark [63] further added in indirect object, i.e., $(subject, predicate, object, indirect object)$. To solve sparsity in these symbolic representations, the following studies learned distributed event representations, i.e., event embeddings, by composing their components. Relatively early studies applied additive composition methods [64], [189], [190]. They added linearly transformed predicate and argument embeddings. Bai et al. [190] further added

the embeddings of event sentences. Then, Weber et al. [191] and Ding et al. [192] proposed tensor-based models to capture multiplicative interactions between event elements. To introduce external knowledge, Ding et al. [192] further learned the intent and sentiment of the event. Recently, other methods were adopted, such as probabilistic models [193] and FCN [63], [194]. Lee and Goldwasser [194] further introduced event discourse relations via their composition to learn relation-aware event embeddings. Besides getting composite event embeddings, Li et al. [65] organized the events into a graph and used GNN to update embeddings.

Script modeling. Early studies modeled event pairs in the script, which ignore event order or only consider limited event order of event pair. Different from PMI used in [183], they obtained the coherence score of event pairs via Bi-gram [188], [195], FCN [63], cosine similarity [191], [193], or translation-based composition [194]. Other studies modeled the whole event chain in the script via language model [196], [197], neural network-based probabilistic model [189], or LSTM [64], [198], [199]. Wang et al. [64] additionally made use of event pair-based models. Different from these single-chain models, Chambers and Jurafsky [55] and Bai et al. [190] aggregated the results from multiple event chains. Other than a chain, Li et al. [65] and Ding et al. [192] organized the events into a graph and modeled it via GNN.

Evaluation. The narrative cloze test in [183] cannot recognize multiple plausible events, since only the original subsequent event is viewed as the answer. Thus, Modi [189] proposed the adversarial narrative cloze test. It asks models to distinguish a correct chain from a negative one, a copy of the correct chain with the subsequent event replaced randomly. The Multiple Choice Narrative Cloze (MCNC) test [63] further restricts the subsequent event to a few choices (see the bottom part of Fig. 9). Based on MCNC, Lee and Goldwasser [193] proposed Multiple Choice Narrative Sequences (MCNS) and Multiple Choice Narrative Explanation (MCNE) to evaluate the ability of inferring long event sequences. MCNS creates candidates for each step. MCNE additionally needs the end event and infers what happened in between. Among these tests, MCNC is widely used.

In a word, as presented in Table 8, there are two ways to represent script. One is entity-centric, organizing the script events into chains according to different participants. The other is the emerging event-centric method, organizing all the script events into a graph. The former is better at modeling the relations between the participant and its script events, while the latter is better at modeling the interactions of script events. Combining their advantages remains to be studied. There are also some other challenges. For example, how to really predict the subsequent script event instead of choosing it from given candidates? How to combine the information from event instances and script events?

5.1.2 Temporal KG Prediction

Temporal KG prediction is to predict events for future timestamps given the historical temporal KG. Formally,

Definition 10. Temporal KG prediction: Given the historical temporal KG in the form of KG sequence with timestamps from 0 to t , i.e., $\mathcal{G} = \{\mathcal{G}_0, \mathcal{G}_1, \dots, \mathcal{G}_t\}$, it predicts events (s, p, o) for future timestamp $t+1$, including $o^* =$

TABLE 8
Comparison of methods for script event prediction.

| Method category | Description | Strengths | Weaknesses |
|-----------------|----------------------------|-----------------------------------|---|
| Chain-modeling | Model event pairs or chain | Good at modeling behavior trends | Miss information from script events of other entities |
| Graph-modeling | Apply GNN models | Catch more comprehensive contexts | Difficult to capture behavior trend of each entity |

$$\arg \max_{\forall o} \Pr(o|s, p, t+1, \mathcal{G}), s^* = \arg \max_{\forall s} \Pr(s|p, o, t+1, \mathcal{G}), \text{ and } p^* = \arg \max_{\forall p} \Pr(p|s, o, t+1, \mathcal{G}).$$

For example,

Example 6. Input: (?, selected as, President of America, 20 January 2009) and historical events before 20 January 2009.

Output: Barack Obama.

Temporal KG prediction requires models to understand historical events. Thus, as shown in Fig. 10, existing methods formalize it into history modeling and future prediction. They model historical events and their evolution. Based on these, they predict the future event and output a distribution over all the events with the peak as the answer. According to the organization and modeling of historical events, methods can be split into graph sequence-based and temporal point process-based ones. The former organizes historical events as a sequence of graphs where each graph contains the events that occurred at the corresponding timestamp. The latter considers historical events as event points.

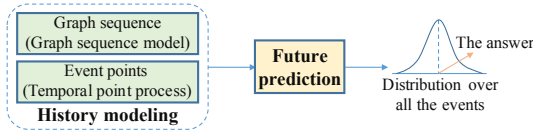


Fig. 10. Existing formalization of temporal KG prediction.

Graph sequence-based methods. In history modeling, they extract related subgraph sequence. They then learn the evolutionary embedding(s) of the graph sequence or subjects, predicates, and objects. GNN and RNN are usually applied for graph and sequence evolution, respectively. Upon the embedding(s), in future prediction, they do argument (i.e., subject/object) prediction (and predicate prediction) via the distribution of all the candidate arguments (or predicates).

Existing methods mainly differ in history extraction. Jin et al. [186] extracted historical events of the given subject and predicate to predict the object. Jin et al. [187] extracted historical events of the given subject. Unlike these heuristic methods, Han et al. [200] started from the given subject and iteratively sampled relevant edges of arguments included in the subgraph. Li et al. [201] and Sun et al. [202] considered history extraction as sequential decision and searched from history via reinforcement learning. Other than the above subgraph sequences, Li et al. [203] used the entire graph sequence at the last several timestamps. These studies model the temporal KG in a discrete-time domain. To encode the continuous dynamics of temporal KG, Ding et al. [204] extended neural ordinary differential equations [205] to GCN.

Temporal point process-based methods. They introduce conditional intensity function, which is powerful for modeling the evolutionary characters of event points, to do history modeling and future prediction in a unified framework.

Traditional methods manually specified the conditional intensity function and modeled events without arguments

[206], [207]. Some recent works [184], [185], [208] extended them to neural methods, using deep neural networks to fit the conditional intensity function. Neural temporal point process-based methods are powerful in considering the semantics of event elements. For example, Trivedi et al. [184] applied RNN to learn the evolutionary embeddings of arguments. They made event argument prediction by estimating the conditional probability of an event based on the evolutionary embeddings of the involved arguments and the predicate embedding. That is, the occurrence of an event was modeled as a multivariate point process, whose conditional intensity function was modulated by its score based on the involved embeddings. This method models events occurring continuously, where no events occur at the same timestamp. To further model concurrent events, Han et al. [208] aggregated the object embeddings in the concurrent events related to the given subject into a hidden vector. Then, they modeled time as a random variable and deployed the Hawkes process [209] on temporal KG to capture the underlying dynamics, where a continuous-time LSTM was used to estimate the intensity function.

In general, existing temporal KG prediction methods (summarized in Table 9) usually focus on events with three arguments (i.e., subject, object, and time) and perform argument or predicate prediction with other elements of future events given. Thus, there are several challenges. For example, how to model historical events with various arguments? How to practically predict future events on the whole?

5.1.3 Other Basic Applications

There are also direct analyses on EKG, such as timeline generation and abductive reasoning. For example, Gottschalk and Demidova [210] generated cross-lingual event timelines using the multilingual event-centric temporal KG EventKG [28]. For a query entity or event, they relied on EventKG to provide information concerning the event popularity and relation strength between events and the query³. Gottschalk and Demidova [211] and Gottschalk and Demidova [212] handled biographical timeline generation. For a query person, they extracted the most relevant biographical data from EventKG based on event popularity, relation strength, and predicate labels⁴. Du et al. [213] proposed an event graph enhanced pretrained language model based on variational autoencoder for abductive reasoning, which finds the most reasonable explainable events for the observed events.

5.2 Deep Applications

EKG can further facilitate many downstream applications, such as search, question-answering, recommendation, financial quantitative investments, and text generation. For

3. See <http://eventkg-timeline.l3s.uni-hannover.de/>

4. See <http://eventkg-biographies.l3s.uni-hannover.de/>

TABLE 9
Comparison of methods for Temporal KG prediction.

| Method category | Description | Strengths | Weaknesses |
|------------------------------|---|--|---|
| Graph sequence-based | Predict events based on related historical subgraph sequences | Capture the semantic information of concurrent entities and temporal order | Ignore precise time information |
| Temporal point process-based | Predict events based on the evolutionary characters of event points | Can model precise time information | High computational complexity and not capable for concurrent events |

TABLE 10
Summarization of applications on EKG.

| Application category | Application description |
|------------------------------------|---|
| Script event prediction | Predict the subsequent script events given the historical scripts (many academic studies) |
| Temporal KG prediction | Predict future events given the historical temporal KG (many academic studies) |
| Timeline generation | Generate event/biographical timelines using EKG [210], [211], [212] |
| Abductive reasoning | Introduce EKG as additional knowledge into abductive reasoning [213] |
| Search | Enable search engines/systems with the searching ability on events based on EKG [1], [2] |
| Question-answering | Query on events based on EKG [3] |
| Recommendation | Incorporate additional knowledge from EKG to improve recommendation [4] |
| Financial quantitative investments | Embeddings from EKG help quantitative trading methods [5] |
| Text generation | Better serve the structured information in EKG in a user-friendly manner [6] |

example, Rudnik et al. [1] developed an event-based search engine⁵ able to query both KGs and news articles. Specifically, they mapped events described in news articles to those in Wikidata [214], and attributes from Wikidata were used to annotate the news articles. They then constructed an event-oriented KG and an event-based search engine. Yang et al. [2] implemented a temporal semantic search system for clinical diagnosis and treatment of traditional Chinese medicine. It consists of the offline and online parts. The former is about the construction, storage, and indexing of the temporal KG, and the latter is about the understanding, conversion, and execution of the search sentences. Souza Costa et al. [3] addressed answering event-centric questions and constructed the first event-centric question-answering dataset based on the event-centric temporal KG EventKG [28]. Wu et al. [4] proposed a GCN-based method for Point-of-Interest (POI) recommendations, which incorporates tourists' behavior patterns obtained from event-centric tourism KG, to capture the relations between users and POIs effectively. Cheng et al. [5] presented a KG-based event embedding framework for financial quantitative investments, where the learned embeddings were fed to downstream quantitative trading methods. A mobile mini-app and a Web-based desktop platform were developed based on this framework, obtaining great accumulated portfolio returns. Colas et al. [6] focused on graph-to-text generation to better serve the structured information in the graph in a user-friendly manner. For each event from EventKG, they augmented the data with additional information from Wikidata and linked the event to a Wikipedia page for text generation.

In general, as summarized in Table 10, there are many basic and deep applications on EKG. Specifically, studies on script event and temporal KG predictions are richer. Since EKG is a relatively new concept, concrete applications and real-world use cases are in a small number. Thus, exploring

more practical applications on EKG is promising.

6 FUTURE DIRECTIONS

There are many researches and achievements on EKG. However, there are still several directions to focus on and investigate further. In this section, we look deep into them.

6.1 High-performance Event Acquisition

Recent event acquisition researches are far from meeting the application requirements in effectiveness and efficiency. Especially, the precision of event and event relation extractions is low. Thus, it hinders the construction of high-quality EKGs. Besides, existing models usually do not pay attention to the complexity problem. However, models of high parameter and time complexity go against the fast construction of EKGs from massive data. Thus, highly effective and efficient event acquisition is an essential future direction.

6.2 Multi-modal Knowledge Processing

Events are presented in texts, images, audios, and videos in the real world. However, existing EKG researches usually process text, ignoring the information in other modalities. Very few studies look into multi-modal event representation learning [215] and extraction [216]. Actually, events from different modalities disambiguate and complement mutually. Thus, jointly using multi-modal information is important. Specifically, events from all modalities should be represented in a unified framework, event acquisition should handle multi-modal extraction, and reasoning on EKG should consider the multi-modal information.

6.3 Interpretable EKG Research

Existing EKG researches mainly apply deep learning methods to fit the training data. However, they usually lack

5. See <https://asrael.eurecom.fr/search-engine-old/home>

interpretability, i.e., there are no definite ideas about why and how they work. Actually, knowing the reasons for the final results is useful for adopting them in real applications. It is friendly and convincing to explain why the final results are the given ones. In the future, interpretable EKG research is an important direction.

6.4 Practical EKG Research

Currently, the related tasks and methods of EKG are far from real-world scenarios. For the related tasks, some task formalizations are idealized. For example, complete a missing element in an existing event, predict a future script event via choosing it from several candidates, and predict an element for a future event. Researches under more practical formalizations are of great importance for applications. For the methods on EKG, GNN is widely used. However, with the lack of EKG datasets and the simplified tasks, these methods employ GNN on simplified datasets of EKG (e.g., [65] and [204]), similar to that on KG. Actually, EKG consists of events, entities, and their relations. Thus, future methods should pay attention to the atomicity of events with arguments, and the relations between events and between entities.

7 CONCLUSIONS

EKG is important for many applications, including search, question-answering, recommendation, financial quantitative investments, and text generation. This paper presents a survey on EKG from different views comprehensively. Specially, we looked deep into the history, ontology, instance, and application views of EKG. Its history, definitions, schema induction, acquisition, related representative graphs/systems, and applications are thoroughly studied. Based on the development trends therein, prospective directions are further summarized for future research on EKG.

ACKNOWLEDGMENTS

The work was supported in part by the Lenovo-CAS Joint Lab Youth Scientist Project, in part by the Foundation and Frontier Research Key Program of Chongqing Science and Technology Commission (No. cstc2017jcyjBX0059), in part by the Youth Innovation Promotion Association CAS under Grant 20144310, in part by the National Natural Science Foundation of China under Grants 62002341, U1911401, 61772501, U1836206, and 91646120, and in part by the GFKJ Innovation Program.

REFERENCES

- [1] C. Rudnik, T. Ehrhart, O. Ferret, D. Teyssou, R. Troncy, and X. Tannier, "Searching news articles using an event knowledge graph leveraged by Wikidata," in *WWW*, 2019, p. 1232–1239.
- [2] C. Yang, W. Li, X. Zhang, R. Zhang, and G. Qi, "A temporal semantic search system for traditional Chinese medicine based on temporal knowledge graphs," in *JIST*, 2019, pp. 13–20.
- [3] T. Souza Costa, S. Gottschalk, and E. Demidova, "Event-QA: A dataset for event-centric question answering over knowledge graphs," in *CIKM*, 2020, p. 3157–3164.
- [4] J. Wu, X. Zhu, C. Zhang, and Z. Hu, "Event-centric tourism knowledge graph—a case study of hainan," in *KSEM*, 2020, pp. 3–15.
- [5] D. Cheng, F. Yang, X. Wang, Y. Zhang, and L. Zhang, "Knowledge graph-based event embedding framework for financial quantitative investments," in *SIGIR*, 2020, p. 2221–2230.
- [6] A. Colas, A. Sadeghian, Y. Wang, and D. Z. Wang, "EventNarrative: A large-scale event-centric dataset for knowledge graph-to-text generation," 2021.
- [7] Z. Zhang, Y. Wu, and Z. Wang, "A survey of open domain event extraction," 2018.
- [8] W. Xiang and B. Wang, "A survey of event extraction from text," *IEEE Access*, vol. 7, pp. 173 111–173 137, 2019.
- [9] Y. Chen, Z. Ding, Q. Zheng, Y. Qin, R. Huang, and N. Shah, "A history and theory of textual event detection and recognition," *IEEE Access*, vol. 8, pp. 201 371–201 392, 2020.
- [10] Q. Li, H. Peng, J. Li, Y. Hei, R. Sun, J. Sheng, S. Guo, L. Wang, J. Wu, A. Beheshti, and P. S. Yu, "A comprehensive survey on schema-based event extraction with deep learning," *arXiv preprint arXiv:2107.02126*, 2021.
- [11] X. Chen and Q. Li, "Event modeling and mining: A long journey toward explainable events," *The VLDB Journal*, vol. 29, no. 1, pp. 459–482, 2020.
- [12] K. Liu, Y. Chen, J. Liu, X. Zuo, and J. Zhao, "Extracting event and their relations from texts: A survey on recent research progress and challenges," *The VLDB Journal*, vol. 1, pp. 22–39, 2020.
- [13] J. Lu and V. Ng, "Event coreference resolution: A survey of two decades of research," in *IJCAI*, 2018, pp. 5479–5486.
- [14] Z. Vendler, "Verbs and times," *The Philosophical Review*, vol. 66, no. 2, pp. 143–160, 1957.
- [15] D. Davidson, "The logical form of action sentences," *Essays on Actions and Events*, pp. 105–148, 1967.
- [16] A. P. Mourelatos, "Events, processes, and states," *Linguistics and Philosophy*, vol. 2, no. 3, pp. 415–434, 1978.
- [17] J. Pustejovsky, "The syntax of event structure," *Cognition*, vol. 41, no. 1, pp. 47–81, 1991.
- [18] R. Grishman and B. Sundheim, "Message Understanding Conference - 6: A brief history," in *COLING*, 1996, pp. 466–471.
- [19] G. Doddington, A. Mitchell, M. Przybicki, L. Ramshaw, S. Strassel, and R. Weischedel, "The Automatic Content Extraction (ACE) program – tasks, data, and evaluation," in *LREC*, 2004, pp. 837–840.
- [20] M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky, "SemEval-2007 task 15: TempEval temporal relation identification," in *the Workshop on Semantic Evaluations*, 2007, p. 75–80.
- [21] P. Bramsen, P. Deshpande, Y. K. Lee, and R. Barzilay, "Inducing temporal graphs," in *EMNLP*, 2006, pp. 189–198.
- [22] O. Kolomyets, S. Bethard, and M.-F. Moens, "Extracting narrative timelines as temporal dependency structures," in *ACL*, 2012, p. 88–97.
- [23] Q. Do, W. Lu, and D. Roth, "Joint inference for event timeline construction," in *EMNLP-CoNLL*, 2012, pp. 677–687.
- [24] G. Glavaš and J. Šnajder, "Event graphs for information retrieval and multi-document summarization," *Expert Systems with Applications*, vol. 41, no. 15, pp. 6904–6916, 2014.
- [25] —, "Construction and evaluation of event graphs," *Natural Language Engineering*, vol. 21, no. 4, pp. 607–652, 2015.
- [26] M. Rospocher, M. van Erp, P. Vossen, A. Fokkens, I. Aldabe, G. Rigau, A. Soroa, T. Ploeger, and T. Bogaard, "Building event-centric knowledge graphs from news," *Journal of Web Semantics*, vol. 37–38, pp. 132–151, 2016.
- [27] Z. Li, S. Zhao, X. Ding, and T. Liu, "EEG: Knowledge base for event evolutionary principles and patterns," in *SMP*, 2017, pp. 40–52.
- [28] S. Gottschalk and E. Demidova, "EventKG: A multilingual event-centric temporal knowledge graph," in *ESWC*, 2018, pp. 272–287.
- [29] X. Ding, Z. Li, T. Liu, and K. Liao, "ELG: An event logic graph," *arXiv preprint arXiv:1907.08015*, 2019.
- [30] C. Walker, S. Strassel, J. Medero, and K. Maeda, "ACE 2005 multilingual training corpus," *Linguistic Data Consortium*, vol. 57, p. 45, 2006.
- [31] R. C. Schank and R. P. Abelson, *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*. Lawrence Erlbaum Associates, 1977.
- [32] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The berkeley FrameNet project," in *COLING-ACL*.
- [33] W. Lehnert, C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland, "University of Massachusetts: MUC-4 test results

- and analysis," in *the 4th Conference on Message Understanding*, 1992, p. 151–158.
- [34] N. Chinchor, D. D. Lewis, and L. Hirschman, "Evaluating message understanding systems: An analysis of the third message understanding conference (MUC-3)," *Computational Linguistics*, vol. 19, no. 3, p. 409–449, 1993.
- [35] D. Freitag, "Toward general-purpose learning for information extraction," in *COLING-ACL*, 1998, p. 404–408.
- [36] H. L. Chieu, H. T. Ng, and Y. K. Lee, "Closing the gap: Learning-based information extraction rivaling knowledge-engineering methods," in *ACL*, 2003, pp. 216–223.
- [37] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen, "Automatic acquisition of domain knowledge for information extraction," in *COLING*, 2000, pp. 940–946.
- [38] M. Surdeanu, J. Turmo, and A. Ageo, "A hybrid approach for the acquisition of information extraction patterns," in *the Workshop on ATEM*, 2006, pp. 48–55.
- [39] S. Patwardhan and E. Riloff, "Effective information extraction with semantic affinity patterns and relevant regions," in *EMNLP-CoNLL*, 2007, pp. 717–727.
- [40] L. Huang and H. Ji, "Semi-supervised new event type induction and event detection," in *EMNLP*, 2020, pp. 718–724.
- [41] K. Sudo, S. Sekine, and R. Grishman, "An improved extraction pattern representation model for automatic IE pattern acquisition," in *ACL*, 2003, pp. 224–231.
- [42] E. Filatova, V. Hatzivassiloglou, and K. McKeown, "Automatic creation of domain templates," in *COLING/ACL*, 2006, pp. 207–214.
- [43] N. Chambers and D. Jurafsky, "Template-based information extraction without the templates," in *ACL-HLT*, 2011, pp. 976–986.
- [44] N. Balasubramanian, S. Soderland, Mausam, and O. Etzioni, "Generating coherent event schemas at scale," in *EMNLP*, 2013, pp. 1721–1731.
- [45] N. Chambers, "Event schema induction with a probabilistic entity-driven model," in *EMNLP*, 2013, pp. 1797–1807.
- [46] K.-H. Nguyen, X. Tannier, O. Ferret, and R. Besançon, "Generative event schema induction with entity disambiguation," in *ACL-IJCNLP*, 2015, pp. 188–197.
- [47] L. Huang, T. Cassidy, X. Feng, H. Ji, C. Voss, J. Han, and A. Sil, "Liberal event extraction and event schema induction," in *ACL*, 2016, pp. 258–268.
- [48] Q. Yuan, X. Ren, W. He, C. Zhang, X. Geng, L. Huang, H. Ji, C.-Y. Lin, and J. Han, "Open-schema event profiling for massive news corpora," 2018, p. 587–596.
- [49] B. QasemiZadeh, M. R. L. Petruck, R. Stodden, L. Kallmeyer, and M. Candito, "SemEval-2019 task 2: Unsupervised lexical frame induction," in *the Workshop on Semantic Evaluation*, 2019, pp. 16–30.
- [50] K. Yamada, R. Sasano, and K. Takeda, "Semantic frame induction using masked word embeddings and two-step clustering," in *ACL-IJCNLP*, 2021, pp. 811–816.
- [51] J. Shen, Y. Zhang, H. Ji, and J. Han, "Corpus-based open-domain event type induction," in *EMNLP*, 2021, pp. 5427–5440.
- [52] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *JMLR*, vol. 3, p. 993–1022, 2003.
- [53] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional Transformers for language understanding," in *NAACL-HLT*, 2019, pp. 4171–4186.
- [54] V. Raskin, S. Nirenburg, C. F. Hempelmann, I. Nirenburg, and K. E. Triezenberg, "The genesis of a script for bankruptcy in ontological semantics," in *HLT-NAACL Workshop on Text Meaning*, 2003, pp. 30–37.
- [55] N. Chambers and D. Jurafsky, "Unsupervised learning of narrative schemas and their participants," in *ACL-IJCNLP*, 2009, pp. 602–610.
- [56] M. Regneri, A. Koller, and M. Pinkal, "Learning script knowledge with Web experiments," in *ACL*, 2010, pp. 979–988.
- [57] J. C. K. Cheung, H. Poon, and L. Vanderwende, "Probabilistic frame induction," in *NAACL-HLT*, 2013, pp. 837–846.
- [58] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [59] J. W. Orr, P. Tadepalli, J. R. Doppa, X. Fern, and T. G. Dietterich, "Learning scripts as hidden markov models," in *AAAI*, 2014, p. 1565–1571.
- [60] N. Weber, R. Rudinger, and B. Van Durme, "Causal inference of script knowledge," in *EMNLP*, 2020, pp. 7583–7596.
- [61] N. Weber, A. Belyy, N. Holzenberger, R. Rudinger, and B. V. Durme, "Schema curation via causal association rule mining," *arXiv preprint arXiv:2104.08811*, 2021.
- [62] M. Ciosici, J. Cummings, M. DeHaven, A. Hedges, Y. Kankanampati, D.-H. Lee, R. Weischedel, and M. Freedman, "Machine-assisted script curation," in *NAACL-HLT*, 2021, pp. 8–17.
- [63] M. Granroth-Wilding and S. Clark, "What happens next? Event prediction using a compositional neural network model," in *AAAI*, 2016, p. 2727–2733.
- [64] Z. Wang, Y. Zhang, and C.-Y. Chang, "Integrating order information and event relation for script event prediction," in *EMNLP*, 2017, pp. 57–67.
- [65] Z. Li, X. Ding, and T. Liu, "Constructing narrative event evolutionary graph for script event prediction," in *IJCAI*, 2018, pp. 4201–4207.
- [66] M. Li, Q. Zeng, Y. Lin, K. Cho, H. Ji, J. May, N. Chambers, and C. Voss, "Connecting the dots: Event graph schema induction with path language modeling," in *EMNLP*, 2020, pp. 684–695.
- [67] M. Li, S. Li, Z. Wang, L. Huang, K. Cho, H. Ji, J. Han, and C. Voss, "Future is not one-dimensional: Graph modeling based complex event schema induction for event prediction," *arXiv preprint arXiv:2104.06344*, 2021.
- [68] D. Ahn, "The stages of event extraction," in *the Workshop on Annotating and Reasoning about Time and Events*, 2006, pp. 1–8.
- [69] Z. Chen and H. Ji, "Language specific issue and feature exploration in Chinese event extraction," in *NAACL-HLT*, 2009, pp. 209–212.
- [70] S. Liao and R. Grishman, "Using document level cross-event inference to improve event extraction," in *ACL*, 2010, pp. 789–797.
- [71] Z. Song, A. Bies, S. Strassel, T. Riese, J. Mott, J. Ellis, J. Wright, S. Kulick, N. Ryant, and X. Ma, "From light to rich ERE: Annotation of entities, relations, and events," in *the Workshop on Events: Definition, Detection, Coreference, and Representation*, 2015, pp. 89–98.
- [72] S. Ebner, P. Xia, R. Culkin, K. Rawlins, and B. Van Durme, "Multi-sentence argument linking," in *ACL*, 2019, pp. 8057–8077.
- [73] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks," in *ACL-IJCNLP*, 2015, pp. 167–176.
- [74] 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*, 2016, pp. 190–203.
- [75] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *NAACL-HLT*, 2018, pp. 2227–2237.
- [76] S. Yang, D. Feng, L. Qiao, Z. Kan, and D. Li, "Exploring pre-trained language models for event extraction and generation," in *ACL*, 2019, pp. 5284–5294.
- [77] S. Deng, N. Zhang, L. Li, H. Chen, H. Tou, M. Chen, F. Huang, and H. Chen, "OntoED: Low-resource event detection with ontology embedding," in *ACL-IJCNLP*, 2021.
- [78] X. Du and C. Cardie, "Event extraction by answering (almost) natural questions," in *EMNLP*, 2020, pp. 671–683.
- [79] J. Liu, Y. Chen, K. Liu, W. Bi, and X. Liu, "Event extraction as machine reading comprehension," in *EMNLP*, 2020, pp. 1641–1651.
- [80] T. H. Nguyen, K. Cho, and R. Grishman, "Joint event extraction via recurrent neural networks," in *NAACL-HLT*, 2016, pp. 300–309.
- [81] 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 *AAAI*, 2018.
- [82] X. Liu, Z. Luo, and H. Huang, "Jointly multiple events extraction via attention-based graph information aggregation," in *EMNLP*, 2018, pp. 1247–1256.
- [83] T. M. Nguyen and T. H. Nguyen, "One for all: Neural joint modeling of entities and events," in *AAAI*, vol. 33, no. 01, 2019, pp. 6851–6858.
- [84] D. Wadden, U. Wennberg, Y. Luan, and H. Hajishirzi, "Entity, relation, and event extraction with contextualized span representations," *arXiv preprint arXiv:1909.03546*, 2019.
- [85] Y. Lin, H. Ji, F. Huang, and L. Wu, "A joint neural model for information extraction with global features," in *ACL*, 2020, pp. 7999–8009.

- © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See <https://www.ieee.org/publications/rights/index.html> for more information. Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on June 15, 2022 at 02:21:33 UTC from IEEE Xplore. Restrictions apply.

- [138] H. Peng, Y. Song, and D. Roth, "Event detection and co-reference with minimal supervision," in *EMNLP*, 2016, pp. 392–402.
- [139] P. K. Choubey and R. Huang, "Automatic data acquisition for event coreference resolution," in *EACL*, 2021, pp. 1185–1196.
- [140] C. A. Bejan and S. M. Harabagiu, "A linguistic resource for discovering event structures and resolving event coreference," in *LREC*, 2008.
- [141] H. Lee, M. Recasens, A. Chang, M. Surdeanu, and D. Jurafsky, "Joint entity and event coreference resolution across documents," in *EMNLP-CoNLL*, 2012, pp. 489–500.
- [142] A. Cybulska and P. Vossen, "'bag of events' approach to event coreference resolution. supervised classification of event templates," *IJCLA*, vol. 6, no. 2, pp. 11–27, 2015.
- [143] S. Krause, F. Xu, H. Uszkoreit, and D. Weissenborn, "Event linking with sentential features from convolutional neural networks," in *CoNLL*, 2016, pp. 239–249.
- [144] P. K. Choubey and R. Huang, "Event coreference resolution by iteratively unfolding inter-dependencies among events," in *EMNLP*, 2017, pp. 2124–2133.
- [145] K. Kenyon-Dean, J. C. K. Cheung, and D. Precup, "Resolving event coreference with supervised representation learning and clustering-oriented regularization," in *the 7th Joint Conference on Lexical and Computational Semantics*, 2018, pp. 1–10.
- [146] S. Barhom, V. Shwartz, A. Eirew, M. Bugert, N. Reimers, and I. Dagan, "Revisiting joint modeling of cross-document entity and event coreference resolution," in *ACL*, 2019, pp. 4179–4189.
- [147] Y. Zeng, X. Jin, S. Guan, J. Guo, and X. Cheng, "Event coreference resolution with their paraphrases and argument-aware embeddings," in *COLING*, 2020, pp. 3084–3094.
- [148] J. Araki and T. Mitamura, "Joint event trigger identification and event coreference resolution with structured perceptron," in *EMNLP*, 2015, pp. 2074–2080.
- [149] J. Lu and V. Ng, "Learning antecedent structures for event coreference resolution," in *ICMLA*, 2017, pp. 113–118.
- [150] —, "Joint learning for event coreference resolution," in *ACL*, 2017, pp. 90–101.
- [151] Y. Lu, H. Lin, J. Tang, X. Han, and L. Sun, "End-to-end neural event coreference resolution," *arXiv preprint arXiv:2009.08153*, 2020.
- [152] J. Lu and V. Ng, "Span-based event coreference resolution," in *AAAI*, vol. 35, no. 15, 2021, pp. 13 489–13 497.
- [153] H. M. Tran, D. Phung, and T. H. Nguyen, "Exploiting document structures and cluster consistencies for event coreference resolution," in *ACL-IJCNLP*, 2021.
- [154] J. Lu and V. Ng, "Constrained multi-task learning for event coreference resolution," in *NAACL-HLT*, 2021, pp. 4504–4514.
- [155] K. McRae, M. J. Spivey-Knowlton, and M. K. Tanenhaus, "Modeling the influence of thematic fit (and other constraints) in on-line sentence comprehension," *Journal of Memory and Language*, vol. 38, no. 3, pp. 283–312, 1998.
- [156] O. Tilk, V. Demberg, A. Sayeed, D. Klakow, and S. Thater, "Event participant modelling with neural networks," in *EMNLP*, 2016, pp. 171–182.
- [157] X. Hong, A. Sayeed, and V. Demberg, "Learning distributed event representations with a multi-task approach," in *the 7th Joint Conference on Lexical and Computational Semantics*, 2018, pp. 11–21.
- [158] A. García-Durán, S. Dumančić, and M. Niepert, "Learning sequence encoders for temporal knowledge graph completion," in *EMNLP*, 2018, pp. 4816–4821.
- [159] J. Leblay and M. W. Chekol, "Deriving validity time in knowledge graph," in *WWW*, 2018, pp. 1771–1776.
- [160] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *NIPS*, 2013, pp. 2787–2795.
- [161] S. S. Dasgupta, S. N. Ray, and P. Talukdar, "HyTE: Hyperplane-based temporally aware knowledge graph embedding," in *EMNLP*, 2018, pp. 2001–2011.
- [162] Y. Xu, S. Sun, Y. Miao, D. Yang, X. Meng, Y. Hu, K. Wang, H. Song, and C. Miao, "Time-aware graph embedding: A temporal smoothness and task-oriented approach," *arXiv preprint arXiv:2007.11164*, 2020.
- [163] T. Lacroix, G. Obozinski, and N. Usunier, "Tensor decompositions for temporal knowledge base completion," in *ICLR*, 2020.
- [164] C. Xu, M. Nanyeri, F. Alkhoury, H. S. Yazdi, and J. Lehmann, "TeRo: A time-aware knowledge graph embedding via temporal rotation," in *COLING*, 2020, pp. 1583–1593.
- [165] J. Wen, J. Li, Y. Mao, S. Chen, and R. Zhang, "On the representation and embedding of knowledge bases beyond binary relations," in *IJCAI*, 2016, pp. 1300–1307.
- [166] R. Zhang, J. Li, J. Mei, and Y. Mao, "Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding," in *WWW*, 2018, pp. 1185–1194.
- [167] Y. Liu, Q. Yao, and Y. Li, "Role-aware modeling for n-ary relational knowledge bases," in *WWW*, 2021, p. 2660–2671.
- [168] —, "Generalizing tensor decomposition for n-ary relational knowledge bases," in *WWW*, 2020, pp. 1104–1114.
- [169] S. Di, Q. Yao, and L. Chen, "Searching to sparsify tensor decomposition for n-ary relational data," in *WWW*, 2021, p. 4043–4054.
- [170] S. Guan, X. Jin, Y. Wang, and X. Cheng, "Link prediction on n-ary relational data," in *WWW*, 2019, p. 583–593.
- [171] B. Fatemi, P. Taslakian, D. Vázquez, and D. Poole, "Knowledge hypergraphs: Extending knowledge graphs beyond binary relations," in *IJCAI*, 2020, pp. 2191–2197.
- [172] P. Rosso, D. Yang, and P. Cudré-Mauroux, "Beyond triplets: Hyper-relational knowledge graph embedding for link prediction," in *WWW*, 2020, pp. 1885–1896.
- [173] S. Guan, X. Jin, J. Guo, Y. Wang, and X. Cheng, "Link prediction on n-ary relational data based on relatedness evaluation," *TKDE*, 2021.
- [174] —, "NeuInfer: Knowledge inference on n-ary facts," in *ACL*, 2020, pp. 6141–6151.
- [175] M. Galkin, P. Trivedi, G. Maheshwari, R. Usbeck, and J. Lehmann, "Message passing for hyper-relational knowledge graphs," in *EMNLP*, 2020, p. 7346–7359.
- [176] Y. Huang, H. Sun, K. Xu, S. Lu, T. Wang, and X. Zhang, "CoRelatE: Learning the correlation in multi-fold relations for knowledge graph embedding," *Knowledge-Based Systems*, vol. 213, p. 106601, 2021.
- [177] Q. Wang, H. Wang, Y. Lyu, and Y. Zhu, "Link prediction on n-ary relational facts: A graph-based approach," in *ACL Findings*, 2021.
- [178] M. D. Ma, J. Sun, M. Yang, K.-H. Huang, N. Wen, S. Singh, R. Han, and N. Peng, "EventPlus: A temporal event understanding pipeline," in *NAACL-HLT*, 2021, pp. 56–65.
- [179] S. Abdollahi, S. Gottschalk, and E. Demidova, "EventKG+Click: A dataset of language-specific event-centric user interaction traces," in *the CLEOPATRA Workshop of ESWC*, 2020.
- [180] S. Gottschalk, E. Kacupaj, S. Abdollahi, D. Alves, G. Amaral, E. Koutsiana, T. Kuculo, D. Major, C. Mello, G. S. Cheema, A. Sittar, Swati, G. Tahmasebzadeh, and G. Thakkar, "OEKG: The open event knowledge graph," in *the CLEOPATRA Workshop of WWW*, 2021, pp. 61–75.
- [181] Q. Ning, B. Zhou, Z. Feng, H. Peng, and D. Roth, "CogCompTime: A tool for understanding time in natural language," in *EMNLP*, 2018, pp. 72–77.
- [182] H. Zhang, X. Liu, H. Pan, Y. Song, and C. W.-K. Leung, "ASER: A large-scale eventuality knowledge graph," in *WWW*, 2020, p. 201–211.
- [183] N. Chambers and D. Jurafsky, "Unsupervised learning of narrative event chains," in *ACL-HLT*, 2008, pp. 789–797.
- [184] R. Trivedi, H. Dai, Y. Wang, and L. Song, "Know-Evolve: Deep temporal reasoning for dynamic knowledge graphs," in *ICML*, 2017, pp. 3462–3471.
- [185] R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha, "DyRep: Learning representations over dynamic graphs," in *ICLR*, 2019.
- [186] W. Jin, C. Zhang, P. Szekely, and X. Ren, "Recurrent event network for reasoning over temporal knowledge graphs," *arXiv preprint arXiv:1904.05530*, 2019.
- [187] W. Jin, M. Qu, X. Jin, and X. Ren, "Recurrent event network: Autoregressive structure inference over temporal knowledge graphs," in *EMNLP*, 2020.
- [188] K. Pichotta and R. Mooney, "Statistical script learning with multi-argument events," in *EACL*, 2014, pp. 220–229.
- [189] A. Modi, "Event embeddings for semantic script modeling," in *CoNLL*, 2016, pp. 75–83.
- [190] L. Bai, S. Guan, J. Guo, Z. Li, X. Jin, and X. Cheng, "Integrating deep event-level and script-level information for script event prediction," in *EMNLP*, 2021.
- [191] N. Weber, N. Balasubramanian, and N. Chambers, "Event representations with tensor-based compositions," in *AAAI*, 2018, pp. 4946–4953.
- [192] X. Ding, K. Liao, T. Liu, Z. Li, and J. Duan, "Event representation learning enhanced with external commonsense knowledge," in *EMNLP-IJCNLP*, 2019, pp. 4894–4903.

- [193] I.-T. Lee and D. Goldwasser, "FEEL: Featured event embedding learning," in *AAAI*, 2018, pp. 4840–4953.
- [194] —, "Multi-relational script learning for discourse relations," in *ACL*, 2019, pp. 4214–4226.
- [195] B. Jans, S. Bethard, I. Vulić, and M. F. Moens, "Skip n-grams and ranking functions for predicting script events," in *EACL*, 2012, pp. 336–344.
- [196] R. Rudinger, P. Rastogi, F. Ferraro, and B. Van Durme, "Script induction as language modeling," in *EMNLP*, 2015, pp. 1681–1686.
- [197] H. Peng, Q. Ning, and D. Roth, "KnowSemLM: A knowledge infused semantic language model," in *CoNLL*, 2019, pp. 550–562.
- [198] K. Pichotta and R. Mooney, "Learning statistical scripts with LSTM recurrent neural networks," pp. 2800–2806, 2016.
- [199] S. Lv, W. Qian, L. Huang, J. Han, and S. Hu, "SAM-Net: Integrating event-level and chain-level attentions to predict what happens next," in *AAAI*, 2019, pp. 6802–6809.
- [200] Z. Han, P. Chen, Y. Ma, and V. Tresp, "Explainable subgraph reasoning for forecasting on temporal knowledge graphs," in *ICLR*, 2021.
- [201] Z. Li, X. Jin, S. Guan, W. Li, J. Guo, Y. Wang, and X. Cheng, "Search from history and reason for future: Two-stage reasoning on temporal knowledge graphs," in *ACL-IJCNLP*, 2021.
- [202] H. Sun, J. Zhong, Y. Ma, Z. Han, and K. He, "TimeTraveler: Reinforcement learning for temporal knowledge graph forecasting," in *EMNLP*, 2021.
- [203] Z. Li, X. Jin, W. Li, S. Guan, J. Guo, H. Shen, Y. Wang, and X. Cheng, "Temporal knowledge graph reasoning based on evolutionary representation learning," in *SIGIR*, 2021.
- [204] Z. Ding, Z. Han, Y. Ma, and V. Tresp, "Temporal knowledge graph forecasting with neural ode," *arXiv preprint arXiv:2101.05151*, 2021.
- [205] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," in *NIPS*, 2018, pp. 6572–6583.
- [206] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes: Volume 1: Elementary theory and methods*. Springer, 2003.
- [207] J. G. Rasmussen, "Temporal point processes: The conditional intensity function," *Lecture Notes*, 2011.
- [208] Z. Han, Y. Ma, Y. Wang, S. Günnemann, and V. Tresp, "Graph Hawkes neural network for forecasting on temporal knowledge graphs," in *AKBC*, 2020.
- [209] A. G. Hawkes, "Spectra of some self-exciting and mutually exciting point processes," *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971.
- [210] S. Gottschalk and E. Demidova, "EventKG+TL: Creating cross-lingual timelines from an event-centric knowledge graph," in *ESWC*, 2018, pp. 164–169.
- [211] —, "EventKG – the hub of event knowledge on the Web – and biographical timeline generation," *Semantic Web*, vol. 10, no. 6, pp. 1039–1070, 2019.
- [212] —, "EventKG+BT: Generation of interactive biography timelines from a knowledge graph," in *ESWC*, 2020, pp. 91–97.
- [213] L. Du, X. Ding, T. Liu, and B. Qin, "Learning event graph knowledge for abductive reasoning," in *ACL-IJCNLP*, 2021, pp. 5181–5190.
- [214] D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledge base," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.
- [215] L. Zhang, D. Zhou, Y. He, and Z. Yang, "MERL: Multimodal event representation learning in heterogeneous embedding spaces," in *AAAI*, 2021, pp. 14420–14427.
- [216] M. Li, A. Zareian, Q. Zeng, S. Whitehead, D. Lu, H. Ji, and S.-F. Chang, "Cross-media structured common space for multimedia event extraction," in *ACL*, 2020, pp. 2557–2568.



Saiping Guan received the PhD degree in Computer Software and Theory in 2019, from the Institute of Computing Technology, Chinese Academy of Sciences, where she is currently an assistant professor. Her current research interests include event knowledge graph, knowledge graph, n-ary relation, etc. She has published papers in prestigious journals and conferences, including TKDE, Knowledge and Information Systems, WWW, ACL, CIKM, etc. She has received the Best Student Paper Award in ICBK (2017).



Xueqi Cheng received the PhD degree in 2006, from the Institute of Computing Technology, Chinese Academy of Sciences (CAS), where he is currently a professor and the director of the CAS Key Laboratory of Network Data Science and Technology. His main research interests include network science, Web search and data mining, big data processing, distributed computing architecture, etc. He has published over 200 papers in reputable journals and conferences. He has won the Best Full Paper Runner-up Award in CIKM (2017), Best Student Paper Award in SIGIR (2012), and Best Paper Award in CIKM (2011). He currently serves on the editorial board for Journal of Computer Science and Technology, Journal of Computer, etc.



Long Bai is a PhD student in the Institute of Computing Technology, Chinese Academy of Sciences. His current research interests include event knowledge graph, knowledge graph, etc. He has published papers in prestigious journals and conferences, including AAAI, EMNLP, etc.



Fujun Zhang is currently pursuing the master degree in the Institute of Computing Technology, Chinese Academy of Sciences. His current research interests include knowledge graph, information extraction, event extraction, etc.



Zixuan Li is a PhD student in the Institute of Computing Technology, Chinese Academy of Sciences. His current research interests include event knowledge graph, temporal reasoning, event reasoning, etc. He has published papers in prestigious conferences, including SIGIR, ACL, etc.



Yutao Zeng received the master degree in Computer Applied Technology in 2018, from the Institute of Computing Technology, Chinese Academy of Sciences. He currently works at Tencent as a researcher of the Platform and Content Group. His research interests include event knowledge graph, event coreference resolution, etc.



Xiaolong Jin received the PhD degree in Computer Science from Hong Kong Baptist University in 2005. He is currently a professor in the Institute of Computing Technology, Chinese Academy of Sciences. His current research interests include knowledge graph, knowledge engineering, social computing, social networks, etc. He has published more than 200 papers in reputable journals and conferences. He has received the Best (Student/Academic) Paper Awards in ICBK (2017), CIT (2015), CCF Big Data (2015), AINA (2007), and ICAMT (2003).



Jiafeng Guo received the PhD degree in Computer Software and Theory in 2009, from the Institute of Computing Technology, Chinese Academy of Sciences (CAS), where he is currently a professor and the vice director of the CAS Key Laboratory of Network Data Science and Technology. He has worked on a number of topics related to Web search and data mining. His current research is focused on representation learning and neural models for information retrieval and filtering. He has won the Best Full Paper Runner-up Award in CIKM (2017), Best Student Paper Award in SIGIR (2012), and Best Paper Award in CIKM (2011).