Connecting the Dots: Event Graph Schema Induction with Path Language Modeling

Manling Li[1], **Qi Zeng**[1], **Ying Lin**[1], **Kyunghyun Cho**[2], **Heng Ji**[1],
Jonathan May[3], **Nathanael Chambers**[4], **Clare Voss**[5]
[1]University of Illinois at Urbana-Champaign
[2]New York University [3]University of Southern California
[4]US Naval Academy [5]US Army Research Laboratory
{manling2,qizeng2,yinglin8,hengji}@illinois.edu,
kyunghyun.cho@nyu.edu, jonmay@isi.edu,
clare.r.voss.civ@mail.mil, nchamber@usna.edu

Abstract

Event schemas can guide our understanding and ability to make predictions with respect to what might happen next. We propose a new *Event Graph Schema*, where two event types are connected through multiple paths involving entities that fill important roles in a coherent story. We then introduce *Path Language Model*, an auto-regressive language model trained on event-event paths, and select salient and coherent paths to probabilistically construct these graph schemas. We design two evaluation metrics, *instance coverage* and *instance coherence*, to evaluate the quality of graph schema induction, by checking when coherent event instances are covered by the schema graph. Intrinsic evaluations show that our approach is highly effective at inducing salient and coherent schemas. Extrinsic evaluations show the induced schema repository provides significant improvement to downstream end-to-end Information Extraction over a state-of-the-art joint neural extraction model, when used as additional global features to unfold instance graphs.[1]

1 Introduction

Existing approaches to automated event extraction retain the overly simplistic assumption that events are atomic occurrences. Understanding events requires knowledge in the form of a repository of abstracted event schemas (complex event templates). Scripts (Schank and Abelson, 1977) encode frequently recurring event sequences, where events are ordered by temporal relation (Chambers and Jurafsky, 2009), causal relation (Mostafazadeh et al., 2016b), or narrative order (Jans et al., 2012). Event schemas have become increasingly important for natural language understanding tasks such as story ending prediction (Mostafazadeh et al., 2016a) and reading comprehension (Kočiský et al., 2018; Ostermann et al., 2019).

Previous schema induction methods mostly ignore uncertainty, re-occurring events and multiple hypotheses, with limited attention to capture complex relations among events, other than temporal or causal relations. Temporal relations exist between almost all events, even those that are not semantically related; while research in identifying causal relations has been hobbled by low inter-annotator agreement (Hong et al., 2016).

In this paper, we hypothesize that two events are connected when their entity arguments are co-referential or semantically related. For example, in Figure 1, (a) and (b) refer to very different event instances, but they both illustrate a typical scenario where a group of people moved from one place to another and then attacked the destination. From many such event instance pairs, we can induce multiple paths connecting a movement event to a related attack event: the person being moved became the attacker, and the weapon or vehicle being moved became the instrument of the attack. Low-level primitive components of event schemas are abundant, and can be part of multiple, sparsely occurring, higher-level graph schemas. We thus propose a new schema representation, *Event Graph Schema*, where *two* event types are connected by such paths containing entity-entity relations. Each node represents an entity type or event type, and each edge represents an entity-entity relation type or the argument role of an entity played in an event.

However, between two event types, there may also be noisy paths that should be excluded from graph schemas. We define the following criteria to select good paths in a graph schema: (1). **Salience**: A good path should appear frequently between two event types; (2). **Coherence**: Multiple paths between the same pair of event types should tell a coherent story, namely they should co-occur fre-

[1]Code and data sets are attached in Appendix, and will be made publicly available after blind review.

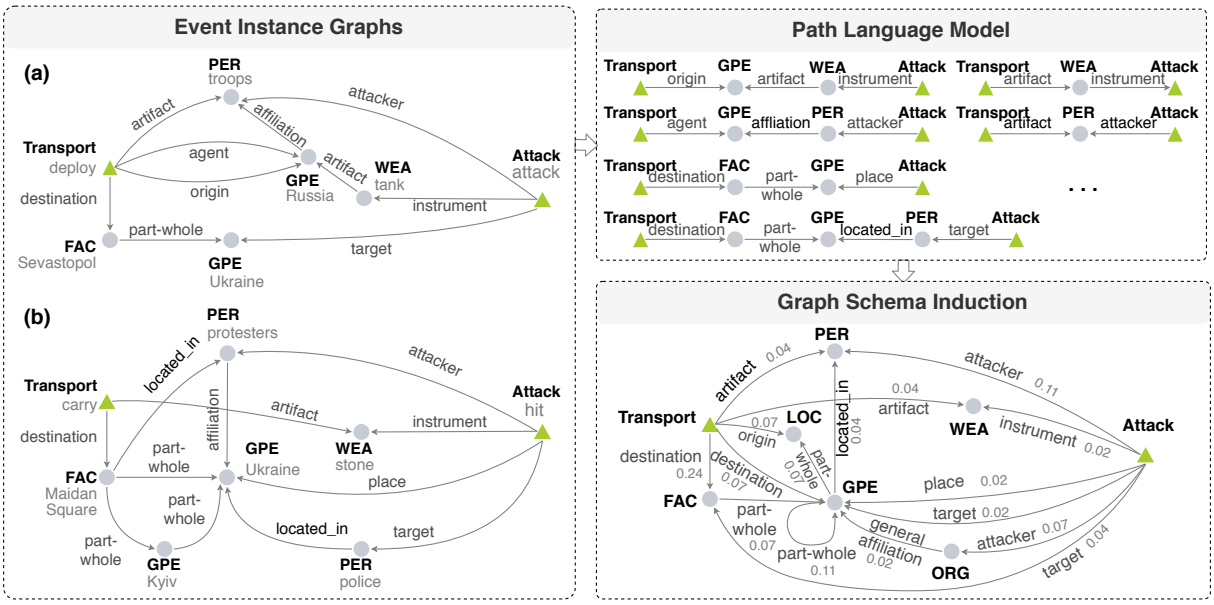

Figure 1: The framework of event graph schema induction. Given a news article, we construct an instance graph for every two event instances from information extraction (IE) results. In this example, instance graph (a) tells the story about Russia deploying troops to attack Ukraine using tanks from Russia; instance graph (b) is about Ukrainian protesters hit police using stones that are being carried to Maidan Square. We learn a path language model to select salient and coherent paths between two event types and merge them into a graph schema. The graph schema between ATTACK and TRANSPORT is an example output containing the top 20% ranked paths.

quently in the same discourse (e.g., the same document). Table 1 shows some examples of good paths and bad paths.

As the first attempt to extract such schemas, we propose a path language model to select paths which clearly indicate how two events are connected through their shared entity arguments or the entity-entity relations between their arguments. For example, in Figure 1 (b), Maidan Square and Ukraine connect events TRANSPORT and ATTACK through the path TRANSPORT $\xrightarrow{\text{DESTINATION}}$ FAC $\xrightarrow{\text{PART-WHOLE}}$ GPE $\xrightarrow{\text{PLACE}^{-1}}$ ATTACK. We train the path language model on two tasks: learning an auto-regressive language model (Ponte and Croft, 1998; Dai and Le, 2015; Peters et al., 2018; Radford et al.; Yang et al., 2019) to predict an edge or a node, given previous edges and nodes in a path, and a neighboring path classification task to predict how likely two paths co-occur. The path language model is trained from all the paths between two event instances from the same document, based on the assumption that events from the same document (especially news document) tell a coherent story.

We propose two intrinsic evaluation metrics, *instance coverage* and *instance coherence*, to assess when event instance graphs are covered by each graph schema, and when different schemas appear in the same document. Intrinsic evaluation on held-out documents demonstrates that our approach can produce highly salient and coherent schemas. We also conduct extrinsic evaluations and show the effectiveness of the induced schema repository in enhancing downstream end-to-end information extraction (IE) tasks, such as entity extraction, relation extraction, event extraction and argument role labeling. In summary, we have the following contributions:

- A novel semantic schema induction framework for the new event schema representation, *Event Graph Schema*, that encodes rich event structures and event-event connections, and two new evaluation metrics to assess graph schemas for coverage and coherence.
- A *Path Language Model* to select salient and coherent event-event paths and construct an event graph schema repository that is probabilistic and semantically coherent.
- The first work to show how to apply event schema to enhance end-to-end IE.

2 Problem Formulation

Given an input document, we extract instances of entities, relations, and events. The type set of enti-

Criteria		Examples	Frequency
Single Path	Salience	**Good** TRANSPORT $\xrightarrow{\text{AGENT}}$ GPE $\xrightarrow{\text{AFFILIATION}^{-1}}$ PER $\xrightarrow{\text{ATTACKER}^{-1}}$ ATTACK	31
		Bad TRANSPORT $\xrightarrow{\text{DESTINATION}}$ GPE $\xrightarrow{\text{AFFILIATION}^{-1}}$ PER $\xrightarrow{\text{ATTACKER}^{-1}}$ ATTACK	2
	Coherence	**Good** TRANSPORT $\xrightarrow{\text{ORIGIN}}$ FAC $\xrightarrow{\text{PART-WHOLE}}$ LOC $\xrightarrow{\text{PART-WHOLE}}$ GPE $\xrightarrow{\text{AFFILIATION}^{-1}}$ PER $\xrightarrow{\text{ATTACKER}^{-1}}$ ATTACK	9
		Bad TRANSPORT $\xrightarrow{\text{AGENT}}$ GPE $\xrightarrow{\text{AFFILIATION}^{-1}}$ PER $\xrightarrow{\text{AFFILIATION}}$ GPE $\xrightarrow{\text{RESIDENT}^{-1}}$ PER $\xrightarrow{\text{TARGET}^{-1}}$ ATTACK	24
Multiple Paths	Coherence	**Good** TRANSPORT $\xrightarrow{\text{DESTINATION}}$ GPE $\xrightarrow{\text{PLACE}^{-1}}$ ATTACK TRANSPORT $\xrightarrow{\text{ARTIFACT}}$ PER $\xrightarrow{\text{LOCATED\_IN}}$ GPE $\xrightarrow{\text{PLACE}^{-1}}$ ATTACK	20
		Bad TRANSPORT $\xrightarrow{\text{DESTINATION}}$ GPE $\xrightarrow{\text{PLACE}^{-1}}$ ATTACK TRANSPORT $\xrightarrow{\text{ORIGIN}}$ GPE $\xrightarrow{\text{PLACE}^{-1}}$ ATTACK	0

Table 1: The criteria of path ranking to construct event schema graph. Frequency is from ACE 2005 annotations. We use '-1' to indicate the reversed edge direction.

ties and events is Φ, and the type set of entity-entity relations and event argument roles is Ψ. For every two event instances, we construct an event instance graph $g = (V, E, \varphi) \in \mathcal{G}$ with all paths connecting the two, as in Figure 1 (a) and (b). V and E are the node and edge sets, and $\varphi : \{V, E\} \rightarrow \{\Phi, \Psi\}$ is a mapping function to obtain the type of each node or edge. Each node $v_i = \langle w_i, \varphi(v_i) \rangle \in V$ represents an entity or an event with text mention w_i, and $\varphi(v_i) \in \Phi$ denotes its node type. Each set of coreferential entities or events is mapped to one single node. Each edge $e_{ij} = \langle v_i, \varphi(e_{ij}), v_j \rangle \in E$ represents an event-argument role or an entity-entity relation, where i and j denote the involved nodes. $\varphi(e_{ij}) \in \Psi$ indicates the edge type. Figure 1 shows two example instance graphs.

Event graph schema induction aims to generate a set of recurring graph schemas $\mathcal{S}$ from instance graphs $\mathcal{G}$. For every event type pair, we induce an event graph schema $s = (U, H) \in \mathcal{S}$, where U and H are the node and edge sets. Each node $u_i = \langle \phi_i \rangle \in U$ is a node type $\phi_i \in \Phi$ in instance graphs $\mathcal{G}$, and each edge $h_{ij} = \langle \phi_i, \psi_{ij}, \phi_j \rangle \in H$ represents an edge type $\psi_{ij} \in \Psi$ in instance graphs $\mathcal{G}$, where ϕ_i and ϕ_j denote the involved node types. Figure 1 shows an example of an induced graph schema between TRANSPORT and ATTACK.

3 Path Language Model based Graph Schema Induction

3.1 Overview

As shown in Table 1, a graph schema for two event types consists of **salient** and **coherent** paths between them. A salient path reveals knowledge of recurring event-event connection patterns. For example, the frequent path in Table 1 shows that the attacker is a member of the government conducting a deployment, which repeatedly appears in the story about attackers sending weapons and people to attack a target place. However, the attacker is unlikely to be affiliated with a target place, so the infrequent path in Table 1 should be excluded from the schema.

In addition, a good path is semantically coherent. For example, the coherent path in Table 1 shows that the origin of transportation is a subarea of the attacker's country, which captures the hierarchical subarea semantics between two places. However, in the bad path example, a person is affiliated with both the origin and destination of the transportation, which is a situation only weakly coherent.

Furthermore, multiple paths in a good schema may be coherent, that is, they co-occur frequently connecting event instance pairs. For example, in Table 1, the destination of transportation is the attack's target, and meanwhile, the people being transported are located there. The co-occurrence of these two paths represents a repetitive pattern to connect TRANSPORT and ATTACK. However, the incoherent example in Table 1 indicates that the attack place is both the destination and the origin of the transportation, where two paths rarely co-occur.

We start by applying Information Extraction (IE) to construct instance graphs between event instances in each document (Section 3.2). We treat a path sequence as a text sequence, and learn an

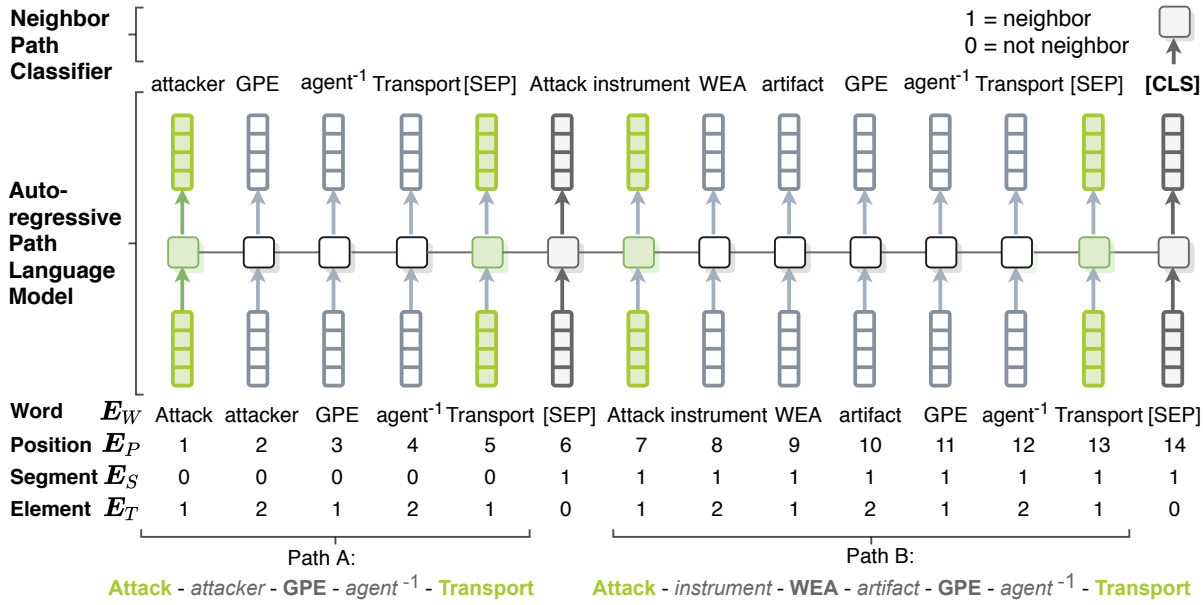

Figure 2: Autoregressive path language model with neighbor path classification.

auto-regressive path language model to score each path (Section 3.3). To capture the coherence between paths, we learn a neighbor path classifier to predict whether two paths co-occur (Section 3.4). The path language model is trained jointly on these two tasks (Section 3.5), which enables us to score and rank paths between event type pairs, and merge salient and coherent paths into graph schemas (Section 3.6).

3.2 Instance Graph Construction

Starting with entities, entity-entity relations, events and their arguments that have been extracted from an input document by IE systems or manual annotation, we construct an event instance graph g for two event instances v and v', that includes all *instance paths* between them. Each instance path

$$p^{\mathbb{I}} = [v, e_{01}, v_1, \ldots, e_{n-1,n}, v']$$

is a sequence of nodes $v, v_1, ..., v' \in V$ and edges $e_{01}, ..., e_{n-1,n} \in E$, such as the instance path *attack* $\xrightarrow{\text{INSTRUMENT}}$ *tank* $\xrightarrow{\text{ARTIFACT}}$ *Russia* $\xrightarrow{\text{AGENT}^{-1}}$ *deploy* in Figure 1 (a). The node instances in each path are distinct to avoid cycles. An event-event path is a sequence of types of nodes and edges,

$$p = [\varphi(v), \varphi(e_{01}), \varphi(v_1), \ldots, \varphi(e_{n-1,n}), \varphi(v')] .$$

For example, the path abstracted from the instance path above is ATTACK $\xrightarrow{\text{INSTRUMENT}}$ WEA $\xrightarrow{\text{ARTIFACT}}$ GPE $\xrightarrow{\text{AGENT}^{-1}}$ TRANSPORT . We consider paths in both directions, i.e., reversed paths are valid.

3.3 Autoregressive Path Language Model

To score and select salient and semantically coherent path sequences, we take a language modeling approach, inspired by node representation learning (Grover and Leskovec, 2016; Goikoetxea et al., 2015) using language model over paths. Autoregressive language model (Ponte and Croft, 1998; Dai and Le, 2015; Peters et al., 2018; Radford et al.; Yang et al., 2019) learns the probability of text sequences as the probability distribution of each word, given its context factorizing the likelihood of prior words into a forward product or, for context in the other direction, a backward product. Similarly, for a path instance $p^{\mathbb{I}}$, we estimate the probability distribution of a node type $\varphi(v_i)$ (or edge type $\varphi(e_{j,j+1})$), given the sequence of previously observed nodes and edges $[\varphi(v), \varphi(e_{01}), \varphi(v_1), \ldots, \varphi(e_{i-1,i})]$, (or $[\varphi(v), \varphi(e_{01}), \varphi(v_1), \ldots, \varphi(v_i)]$), i.e.,

$$\mathcal{L}_{LM} = \sum_{p^{\mathbb{I}}} \Big[\sum_{v_i \in p^{\mathbb{I}}} \log P(\varphi(v_i) | \varphi(v), ..., \varphi(e_{i-1,i})) + \sum_{e_{j,j+1} \in p^{\mathbb{I}}} \log P(\varphi(e_{j,j+1}) | \varphi(v), \varphi(e_{01}), ..., \varphi(v_i)) \Big] .$$

Following (Yang et al., 2019), we apply the Transformer (Vaswani et al., 2017) to learn the probability distribution, with permutation operation (Yang et al., 2019) to capture bidirectional contexts. Unlike in text sequences, we have nodes and edges that alternate within path sequences. As shown in

Figure 2, to distinguish nodes and edges, we add type embedding $\boldsymbol{E}_T = [1, 2, 1, \ldots, 2, 1]$ into the token representation, where 1 stands for nodes, 2 for edges, and 0 for special tokens such as [CLS].

We hypothesize that event instances from the same discourse (e.g., a news document) describe a coherent story, and so we use the paths between them as training paths.

3.4 Neighbor Path Classification

To capture coherence across paths, we train a binary neighbor path classifier to learn the occurrence probability of two paths. For each path $p_i \in \mathcal{P}_{\langle v,v' \rangle}$ between two event instances v and v', we obtain its *neighbor path* set as its co-occuring paths between the same event instances v and v',

$$\mathcal{N}_{p_i} = \{p_j | p_j \in \mathcal{P}_{\langle v,v' \rangle}, v, v' \in V\}.$$

We sample negative neighbor paths from paths that appear between the same event types $\varphi(v)$ and $\varphi(v')$, but never occur with p_i in the corpus.

$$\mathcal{N}'_{p_i} = \{p_j | p_j \in \mathcal{P}_{\langle \varphi(v),\varphi(v') \rangle}, p_j \notin \mathcal{N}_{p_i}\}.$$

The classifier (top of Figure 2) is a linear layer with the classification token $\boldsymbol{x}_{[\text{CLS}]}$ as input,

$$P(p_j \in \mathcal{N}_{p_i}) = \text{sigmoid}\left(\boldsymbol{W}\boldsymbol{x}_{[\text{CLS}]} + \boldsymbol{b}\right).$$

We balance the positive and negative path pairs during training, and optimize cross-entropy loss,

$$\mathcal{L}_{NP} = \sum_{p_i} \Big[\sum_{p_j \in \mathcal{N}_{p_i}} \log P(p_j \in \mathcal{N}_{p_i}) + \sum_{p_j \in \mathcal{N}'_{p_i}} \log(1 - P(p_j \in \mathcal{N}_{p_i})) \Big].$$

3.5 Joint Training

We jointly optimize autoregressive language model loss and neighbor path classifier loss,

$$\mathcal{L} = \mathcal{L}_{LM} + \lambda \mathcal{L}_{NP}.$$

3.6 Graph Schema Construction

Given two event types ϕ and ϕ', we construct a graph schema s by merging the top k percent ranked paths. Paths in $\mathcal{P}_{\langle \phi,\phi' \rangle}$ are ranked in terms of a score function $f(p)$,

$$f(p_i) = f_{\text{LM}}(p_i) + \alpha f_{\text{NP}}(p_i),$$

where $f_{\text{LM}}(p)$ captures salience and coherence of a single path,

$$f_{\text{LM}}(p_i) = \log P([\phi, \psi_{01}, \phi_1, \psi_{12}, ..., \phi']),$$

and where $f_{\text{NP}}(p)$ scores a path p_i by its average probability of co-occuring with other paths $p_j \in \mathcal{P}_{\langle \phi,\phi' \rangle}$ between the given event types ϕ and ϕ',

$$f_{\text{NP}}(p_i) = \frac{1}{|\mathcal{P}_{\langle \phi,\phi' \rangle}|} \sum_{p_j \in \mathcal{P}_{\langle \phi,\phi' \rangle}} \log P(p_j \in \mathcal{N}_{p_i}).$$

We merge instance paths into a graph schema s by mapping nodes of the same type into a single node. We allow some self-loops in the graph, such as GPE $\xrightarrow{\text{PART-WHOLE}}$ GPE. Each path in the schema has a probability,

$$P(p_i) = \frac{\exp(f(p_i))}{\sum_{p_j \in s} \exp(f(p_j))}.$$

Each edge and node is assigned a salience score by aggregating the scores of paths passing through it,

$$f(\psi_{ij}) = \sum_{p \in \{p | \psi_{ij} \in p, p \in s\}} P(p), \quad f(\phi_i) = \sum_{p \in \{p | \phi_i \in p, p \in s\}} P(p).$$

4 Evaluation Benchmark

4.1 Dataset

We use Automatic Content Extraction (ACE) 2005 dataset[2], the widely used dataset with annotated instances of 7 entity types, 6 relation types, 33 event types, and 22 argument roles. We follow the most recent work on ACE IE (Lin et al., 2020) to split the data. We consider the training set as historical data to train the LM, and the test set as our target data to induce schema for target scenarios. The instance graphs of the target data set are constructed from manual annotations. For historical data, we construct event instance graphs from both manual annotations ($\text{Historical}_{\text{ann}}$) and system extraction results ($\text{Historical}_{\text{sys}}$) from the state-of-the-art IE model (Lin et al., 2020). We perform cross-document entity coreference resolution by applying an entity linker (Pan et al., 2017) for both annotated and system generated instance graphs. Table 2 shows the data statistics.

Split	#Docs	#Entities	#Rels	#Events	#Args
$\text{Historical}_{\text{ann}}$	529	47,525	7,152	4,419	7,888
$\text{Historical}_{\text{sys}}$	529	48,664	7,018	4,426	6,614
Validation	40	3,422	728	468	938
Target	30	3,673	802	424	897

Table 2: Data statistics.

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

4.2 Instance Coverage

A salient schema can serve as a skeleton to recover instance graphs. Therefore, we use each graph schema $s \in \mathcal{S}$ to match back to each ground-truth instance graph $g \in \mathcal{G}$ and evaluate their intersection $g \cap s$ in terms of Precision and Recall.

Intersection is obtained by searching instance graphs with each graph schema as a query. Since instance graphs can be regarded as *partially* instantiated graph schema, we employ the substructures of the schema graph, i.e., paths of different lengths, as queries. For example, a path of length $l = 3$ is a triple in graph schema $\langle \phi_i, \psi_{ij}, \phi_j \rangle \in s$. We consider an instance triple $\langle v_m, e_{mn}, v_n \rangle \in g$ matched if instance types match, i.e., $\varphi(v_m){=}\phi_i$, $\varphi(e_{mn}){=}\psi_{ij}$, $\varphi(v_n){=}\phi_j$. Let $|\cdot|_{\mathbb{I}}$ denote the number of instance substructures matched, and $|\cdot|_{\mathbb{S}}$ is the number of schema substructures matched, i.e.,

$$|g \cap s|_{\mathbb{I}} = \sum_{\langle \phi_i, \psi_{ij}, \phi_j \rangle \in s} \text{count}(\langle v_m, e_{mn}, v_n \rangle),$$

$$|g \cap s|_{\mathbb{S}} = \sum_{\langle v_m, e_{mn}, v_n \rangle \in g} \text{count}(\langle \phi_i, \psi_{ij}, \phi_j \rangle).$$

The cardinality for an instance graph and a schema will be the number of substructures in each, i.e.,

$$|g|_{\mathbb{I}} = \sum_{\langle v_m, e_{mn}, v_n \rangle \in g} \text{count}(\langle v_m, e_{mn}, v_n \rangle),$$

$$|s|_{\mathbb{S}} = \sum_{\langle \phi_i, \psi_{ij}, \phi_j \rangle \in s} \text{count}(\langle \phi_i, \psi_{ij}, \phi_j \rangle).$$

By extension, each path of length $l{=}5$ in a graph schema $[\phi_i, \psi_{ij}, \phi_j, \psi_{jk}, \phi_k]$ contains two consecutive triples $\langle \phi_i, \psi_{ij}, \phi_j \rangle, \langle \phi_j, \psi_{jk}, \phi_k \rangle {\in} s$, and a matched instance path contains two consecutive instance triples $\langle v_m, e_{mn}, v_n \rangle, \langle v_n, e_{no}, v_o \rangle {\in} g$, where $\varphi(v_m){=}\phi_i$, $\varphi(e_{mn}){=}\psi_{ij}$, $\varphi(v_n){=}\phi_j$, $\varphi(e_{no}){=}\psi_{jk}$, $\varphi(v_o){=}\phi_k$. Similarly, a path of length $l{=}7$ contains three consecutive triples. Then we compute:

$$\text{Precision} = \frac{\sum_{s \in \mathcal{S}} \sum_{g \in \mathcal{G}} |g \cap s|_{\mathbb{S}}}{\sum_{s \in \mathcal{S}} |s|_{\mathbb{S}}},$$

$$\text{Recall} = \frac{\sum_{s \in \mathcal{S}} \sum_{g \in \mathcal{G}} |g \cap s|_{\mathbb{I}}}{\sum_{g \in \mathcal{G}} |g|_{\mathbb{I}}}.$$

4.3 Instance Coherence

For an instance graph between two events v and v', we hypothesize that the graph is coherent if v and v' are from the same discourse (document). We carefully select 24 documents with each document talking about a unique complex event such as *Iraq War* or *North Korea Nuclear Test*. A coherent schema should have the maximal number of matched instance graphs $g \cap s$ from a single document, but the minimal number of matched graphs connecting two event instances from different documents. We define **Instance Coherence** as the proportion of event-event path instances in graphs within one document.

$$\text{Coherence} = \frac{\sum_{s \in \mathcal{S}} \sum_{g \in \mathcal{G}} \sum_{p \in g \cap s} f(p) \cdot \mathbb{I}_g}{\sum_{s \in \mathcal{S}} \sum_{g \in \mathcal{G}} \sum_{p \in g \cap s} f(p)},$$

where $\mathbb{I}_g$ is an indicator function taking value 1 when g is between event instances from the same document, and value 0 otherwise.

4.4 Schema-Guided Information Extraction

As a case study for extrinsic evaluation, we evaluate the impact of our induced schema[3] on end-to-end Information Extraction (IE). We choose the aforementioned publicly available IE system OneIE (Lin et al., 2020) as our baseline for two reasons: (1) it achieves state-of-the-art results on all IE components; (2) it can easily incorporate global features during decoding by converting each input sentence into an instance graph. Specifically, we take each path in the graph schema as a single global feature, whose weight is automatically learned by the OneIE model during training. Given a candidate graph, OneIE calculates a score for each path feature and adds it into the global score of the graph. In this way, it can promote candidate graphs containing positive global features even if they may have lower local scores.

5 Experiments

5.1 Settings

Baselines. Since we are the first to induce event graph schema, we compare our method to various other path ranking methods: (1) **Frequency Ranking Model** will rank paths between every two event types by the number of associated instance paths in the historical and target data. (2) **Unigram, Bigram, and Trigram Language Models** assign probabilities to path sequences by estimating the probability of each node (or edge) from the unigram, bigram, and trigram frequency counts, respectively. We also include a variant of PathLM by removing the neighbor path classifier (CLS_{NP}) as an ablation study.

[3]The schema is induced from annotated instance graphs of historical data, which is the training data of IE system.

Historical Instance Graphs	Model	Schema@10									Schema@20								
		$l=3$			$l=5$			$l=7$			$l=3$			$l=5$			$l=7$		
		P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
Historical$_{ann}$	**Frequency**	100	37.5	54.6	90.5	48.3	63.0	76.7	9.5	16.9	100	42.6	59.7	87.6	70.6	78.2	63.6	17.9	28.0
	Unigram LM	100	33.7	50.4	87.1	35.4	50.3	63.9	7.3	13.1	100	43.8	60.9	86.0	60.8	71.2	55.4	14.8	23.4
	Bigram LM	100	33.4	50.1	92.6	36.8	52.6	75.4	8.5	15.3	100	43.2	60.3	88.1	63.2	73.6	62.6	16.4	26.0
	Trigram LM	100	39.9	57.0	89.4	41.6	56.7	62.7	8.5	15.0	100	44.6	61.6	85.6	68.2	75.9	53.4	17.8	26.7
	PathLM	100	41.8	**58.9**	83.7	63.8	**72.4**	54.3	16.6	**25.4**	100	44.7	**61.8**	83.0	80.0	81.5	53.8	27.2	**36.1**
	w/o CLS_{NP}	100	39.8	56.9	90.3	58.3	70.9	71.2	14.5	24.1	100	42.9	60.1	85.7	80.1	**82.8**	57.8	25.8	35.6
Historical$_{sys}$	**Frequency**	100	37.6	54.7	87.0	49.4	63.0	68.6	9.8	17.1	100	41.6	58.8	88.5	70.1	78.2	67.8	19.3	29.9
	Unigram LM	100	41.0	58.2	83.7	36.2	50.5	54.3	7.5	13.1	100	44.6	61.7	83.0	66.4	73.8	52.4	17.9	26.7
	Bigram LM	100	39.2	56.3	88.5	37.7	52.8	61.4	7.9	13.9	100	43.5	60.6	86.5	63.8	73.4	58.3	15.3	24.2
	Trigram LM	100	37.3	54.4	89.6	46.8	61.5	65.2	9.8	17.1	100	44.1	61.2	86.2	68.7	76.5	54.5	17.6	26.6
	PathLM	100	41.7	**58.8**	83.2	68.0	**74.8**	51.8	18.5	**27.3**	100	44.8	**61.9**	81.7	85.4	**83.5**	49.6	29.3	**36.9**
	w/o CLS_{NP}	100	40.1	57.3	89.5	55.1	68.2	72.7	14.4	24.1	100	44.7	61.7	83.8	75.9	80.0	54.8	24.7	34.0

Table 3: Instance coverage (%) by checking the intersection of schemas and instance graphs.

Historical	Model	Schema@10	Schema@20
Historical$_{ann}$	**Frequency**	67.8	65.6
	Unigram LM	62.4	69.9
	Bigram LM	59.0	67.5
	Trigram LM	56.6	64.9
	PathLM	**76.0**	**79.9**
	w/o CLS_{NP}	75.3	79.2
Historical$_{sys}$	**Frequency**	60.1	65.6
	Unigram LM	61.8	70.0
	Bigram LM	59.7	69.6
	Trigram LM	55.8	65.8
	PathLM	**76.4**	**78.5**
	w/o CLS_{NP}	73.9	77.1

Table 4: Instance coherence (%) of schema graphs covering top k percent paths, $k = 10, 20$.

Schema@K. To compare the ranking of paths with baselines, we evaluate graph schemas containing top k % ranked paths.

Implementation Details. We use the same hyperparameters as XLNet-base-cased (Yang et al., 2019), with dropout $= 0.5$. $\lambda = 0.1$, and $\alpha = 0.3$. Detailed parameter settings are in Appendix.

5.2 Results and Analysis

We induce 124 and 197 graph schemas for Schema@10 and Schema@20 respectively. Figure 1 shows an output graph schema.[4] From Table 3 and Table 4 we see that PathLM achieves significant improvement on both instance coverage and instance coherence. We make the following observations:

(1) PathLM achieves larger gains compared to baselines on Schema@10 than Schema@20 in Table 3, demonstrating the effectiveness of our ranking approach, especially on top ranked ones.

(2) The improvement relative to baselines on longer path queries (e.g. $l = 7$) is greater than shorter paths (e.g., $l = 3$) in Table 3, showing that our approach is able to capture complex graph structures involving long distance between related events.

(3) The neighbor path classification proves to be effective in enhancing the salience (see 'w/o CLS_{NP}' in Table 3) and coherence (see 'w/o CLS_{NP}' in Table 4) of the induced schemas, showing that salient substructures can be better captured by frequently co-occurring paths.

(4) The schemas induced from Historical$_{sys}$ and Historical$_{ann}$ have comparable performance. This proves our approach is robust to extraction noise and effective even with lower quality input.

Model	Entity	Rel	Event			
			Trig-I	Trig-C	Arg-I	Arg-C
OneIE Baseline	**90.3**	44.7	75.8	72.7	57.8	55.5
+PathLM	90.2	**60.9**	76.0	**73.4**	59.0	**56.6**
w/o CLS_{NP}	90.1	60.3	75.7	72.8	58.3	55.8

Table 5: F_1 score (%) of schema-guided information extraction, including entity extraction (Entity), relation extraction (Rel), event trigger identification (Trig-I) and classification (Trig-C), event argument identification (Arg-I) and argument role classification (Arg-C).

[4] Visualization of schema repository is in Appendix.

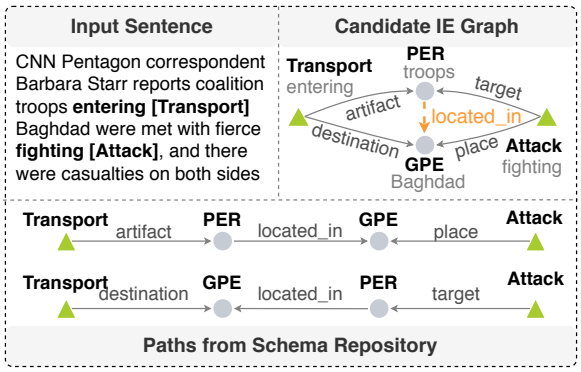

Figure 3: An example showing how schema improves the quality of IE by promoting the candidate IE graph matching paths from schema.

As shown in Table 5, our event graph schemas have provided significant improvement on relation extraction and event extraction which require knowledge of complex connections among events and entities. For example, when decoding candidate IE graph in Figure 3, the LOCATED_IN relation is extracted by promoting the structures matching paths in the graph schema.

5.3 Remaining Challenges

A major challenge in schema induction is to automatically decide the type granularity. For example, if two events happen on the same *street*, it is likely that they are related; if it is a *country* that connects to two events through place arguments, they can be independent. In this case, the fine-grained type information of shared place argument is required in schemas. However, to induce schemas about *war*, geopolitical entities of different granularities should be generalized as GPE.

6 Related Work

Atomic Event Schema Induction. Atomic event schema induction methods (Chambers, 2013; Cheung et al., 2013; Nguyen et al., 2015; Huang et al., 2016; Sha et al., 2016; Yuan et al., 2018) focus on discovering event types and argument roles of individual atomic events.

Narrative Event Schema Induction. Previous work (Chambers and Jurafsky, 2008, 2009, 2010; Jans et al., 2012; Balasubramanian et al., 2013; Pichotta and Mooney, 2014, 2016; Rudinger et al., 2015; Granroth-Wilding and Clark, 2016; Modi, 2016; Mostafazadeh et al., 2016a; Peng et al., 2019) focuses on inducing *narrative schemas* as partially ordered sets of events (represented as verbs) sharing a common argument. The event order is further extended to include causality (Mostafazadeh et al., 2016b; Kalm et al., 2019), and *temporal script graph* is proposed where events and arguments are abstracted as event types and participant types (Modi et al., 2017; Wanzare et al., 2017; Zhai et al., 2019). In our work, we propose a new event graph schema representation to capture more complex connections between events, and use event types instead of verbs as in previous work for more abstraction power.

Path-based Language Model. Language models (LMs) (Ponte and Croft, 1998) achieve great advances on contextualizing LMs in the last few years (Peters et al., 2018; Devlin et al., 2019; Yang et al., 2019). LM has been used over paths to learn node representations in a network (Goikoetxea et al., 2015; Grover and Leskovec, 2016; Dong et al., 2017). To the best of our knowledge, there has not been an effort to incorporate latent linguistic structures into language models based on typed event-event paths. This is also the first work to demonstrate how to leverage event schemas to enhance the performance of an IE system.

Graph Pattern Mining. Motif finding on heterogeneous networks (Prakash et al., 2004; Carranza et al., 2018; Rossi et al., 2019; Hu et al., 2019) discovers highly recurrent instance graph patterns, but fails in abstracting schema graphs to the type level. Previous work applies graph summarization to discover frequent subgraph patterns for heterogeneous networks (Cook and Holder, 1993; Buehrer and Chellapilla, 2008; Li and Lin, 2009; Zhang et al., 2010; Koutra et al., 2014; Wu et al., 2014; Song et al., 2018; Bariatti et al., 2020), but ignores semantic coherence among multiple patterns.

7 Conclusions and Future Work

We propose Event Graph Schema induction as a new step towards semantic understanding of inter-event connections. We develop a path language model based method to construct graph schemas containing salient and semantically coherent event-event paths, which also show effectiveness in enhancing end-to-end IE. In the future, we aim to extend graph schemas to encode hierarchical and temporal relations, as well as rich ontologies in open domain. We will also assemble our graph schemas to represent more complex scenarios involving multiple events, so they can be applied to more downstream applications including event graph completion and event prediction.

Acknowledgement

This research is based upon work supported in part by U.S. DARPA KAIROS Program No. FA8750-19-2-1004, U.S. DARPA AIDA Program No. FA8750-18-2-0014 and Air Force No. FA8650-17-C-7715. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Niranjan Balasubramanian, Stephen Soderland, Oren Etzioni, et al. 2013. Generating coherent event schemas at scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1721–1731.

Francesco Bariatti, Peggy Cellier, and Sébastien Ferré. 2020. Graphmdl: Graph pattern selection based on minimum description length. In *International Symposium on Intelligent Data Analysis*, pages 54–66. Springer.

Gregory Buehrer and Kumar Chellapilla. 2008. A scalable pattern mining approach to web graph compression with communities. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 95–106.

Aldo G. Carranza, Ryan A. Rossi, Anup Rao, and Eunyee Koh. 2018. Higher-order spectral clustering for heterogeneous graphs. *CoRR*, abs/1810.02959.

Nathanael Chambers. 2013. Event schema induction with a probabilistic entity-driven model. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP2013)*, volume 13, pages 1797–1807.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of the 2008 Annual Meeting of the Association for Computational Linguistics (ACL2008)*, pages 789–797.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing (ACL-IJCNLP2009)*.

Nathanael Chambers and Daniel Jurafsky. 2010. A database of narrative schemas. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC2010)*.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic frame induction. *NAACL HLT 2013*, pages 837–846.

Diane J Cook and Lawrence B Holder. 1993. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research*, 1:231–255.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, pages 135–144. ACM.

Josu Goikoetxea, Aitor Soroa, and Eneko Agirre. 2015. Random walks and neural network language models on knowledge bases. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1434–1439. The Association for Computational Linguistics.

Mark Granroth-Wilding and Stephen Clark. 2016. What happens next? event prediction using a compositional neural network model. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 855–864. ACM.

Yu Hong, Tongtao Zhang, Tim O'Gorman, Sharone Horowit-Hendler, Heng Ji, and Martha Palmer. 2016. Building a cross-document event-event relation corpus. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 1–6, Berlin, Germany. Association for Computational Linguistics.

Jiafeng Hu, Reynold Cheng, Kevin Chen-Chuan Chang, Aravind Sankar, Yixiang Fang, and Brian Y. H. Lam. 2019. Discovering maximal motif cliques in large heterogeneous information networks.

In *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*, pages 746–757. IEEE.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R. Voss, Jiawei Han, and Avirup Sil. 2016. Liberal event extraction and event schema induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL2016)*.

Bram Jans, Steven Bethard, Ivan Vulić, and Marie Francine Moens. 2012. Skip n-grams and ranking functions for predicting script events. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 336–344. Association for Computational Linguistics.

Pavlina Kalm, Michael Regan, and William Croft. 2019. Event structure representation: Between verbs and argument structure constructions. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 100–109.

Danai Koutra, U Kang, Jilles Vreeken, and Christos Faloutsos. 2014. Vog: Summarizing and understanding large graphs. In *Proceedings of the 2014 SIAM international conference on data mining*, pages 91–99. SIAM.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Cheng-Te Li and Shou-De Lin. 2009. Egocentric information abstraction for heterogeneous social networks. In *2009 International Conference on Advances in Social Network Analysis and Mining*, pages 255–260. IEEE.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint end-to-end neural model for information extraction with global features. In *Proceedings of the 2020 Annual Meeting of the Association for Computational Linguistics (ACL2020)*.

Ashutosh Modi. 2016. Event embeddings for semantic script modeling. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 75–83.

Ashutosh Modi, Tatjana Anikina, Simon Ostermann, and Manfred Pinkal. 2017. Inscript: Narrative texts annotated with script information. *arXiv preprint arXiv:1703.05260*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016a. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of NAACL-HLT*, pages 839–849.

Nasrin Mostafazadeh, Alyson Grealish, Nathanael Chambers, James Allen, and Lucy Vanderwende. 2016b. Caters: Causal and temporal relation scheme for semantic annotation of event structures. In *Proceedings of the Fourth Workshop on Events*, pages 51–61.

Kiem-Hieu Nguyen, Xavier Tannier, Olivier Ferret, and Romaric Besançon. 2015. Generative event schema induction with entity disambiguation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 188–197.

Simon Ostermann, Michael Roth, and Manfred Pinkal. 2019. Mcscript2. 0: A machine comprehension corpus focused on script events and participants. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (* SEM 2019)*, pages 103–117.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958.

Haoruo Peng, Qiang Ning, and Dan Roth. 2019. Knowsemlm: A knowledge infused semantic language model. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 550–562.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Karl Pichotta and Raymond Mooney. 2014. Statistical script learning with multi-argument events. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 220–229.

Karl Pichotta and Raymond J Mooney. 2016. Learning statistical scripts with lstm recurrent neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Jay M Ponte and W Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281.

Amol Prakash, Mathieu Blanchette, Saurabh Sinha, and Martin Tompa. 2004. Motif discovery in heterogeneous sequence data. In *Biocomputing 2004, Proceedings of the Pacific Symposium, Hawaii, USA, 6-10 January 2004*, pages 348–359. World Scientific.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training.

Ryan A. Rossi, Nesreen K. Ahmed, Aldo G. Carranza, David Arbour, Anup Rao, Sungchul Kim, and Eunyee Koh. 2019. Heterogeneous network motifs. *CoRR*, abs/1901.10026.

Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. 2015. Script induction as language modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1681–1686.

Roger C Schank and Robert P Abelson. 1977. Scripts, plans, goals and understanding: An inquiry into human knowledge structures. *Mhwah, NJ (US): Lawrence Erlbaum Associates*.

Lei Sha, Sujian Li, Baobao Chang, and Zhifang Sui. 2016. Joint learning templates and slots for event schema induction. In *Proceedings of NAACL-HLT*, pages 428–434.

Qi Song, Yinghui Wu, Peng Lin, Luna Xin Dong, and Hui Sun. 2018. Mining summaries for knowledge graph search. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1887–1900.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Lilian Wanzare, Alessandra Zarcone, Stefan Thater, and Manfred Pinkal. 2017. Inducing script structure from crowdsourced event descriptions via semi-supervised clustering. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 1–11.

Ye Wu, Zhinong Zhong, Wei Xiong, and Ning Jing. 2014. Graph summarization for attributed graphs. In *2014 International Conference on Information Science, Electronics and Electrical Engineering*, volume 1, pages 503–507. IEEE.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 5754–5764.

Quan Yuan, Xiang Ren, Wenqi He, Chao Zhang, Xinhe Geng, Lifu Huang, Heng Ji, Chin-Yew Lin, and Jiawei Han. 2018. Open-schema event profiling for massive news corpora. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 587–596.

Fangzhou Zhai, Vera Demberg, Pavel Shkadzko, Wei Shi, and Asad Sayeed. 2019. A hybrid model for globally coherent story generation. In *Proceedings of the Second Workshop on Storytelling*, pages 34–45.

Ning Zhang, Yuanyuan Tian, and Jignesh M Patel. 2010. Discovery-driven graph summarization. In *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, pages 880–891. IEEE.