

# Abstract Meaning Representation Guided Graph Encoding and Decoding for Joint Information Extraction

**Zixuan Zhang and Heng Ji**  
Computer Science Department  
University of Illinois at Urbana-Champaign  
{zixuan11, hengji}@illinois.edu

## Abstract

The tasks of Rich Semantic Parsing, such as Abstract Meaning Representation (AMR), share similar goals with Information Extraction (IE) to convert natural language texts into structured semantic representations. To take advantage of such similarity, we propose a novel AMR-guided framework for joint information extraction to discover entities, relations, and events with the help of a pre-trained AMR parser. Our framework consists of two novel components: 1) an AMR based semantic graph aggregator to let the candidate entity and event trigger nodes collect neighborhood information from AMR graph for passing message among related knowledge elements; 2) an AMR guided graph decoder to extract knowledge elements based on the order decided by the hierarchical structures in AMR. Experiments on multiple datasets have shown that the AMR graph encoder and decoder have provided significant gains and our approach has achieved new state-of-the-art performance on all IE subtasks<sup>1</sup>.

# 1 Introduction

Information extraction (IE) aims to extract structured knowledge as an *information network* (Li et al., 2014) from unstructured natural language texts, while semantic parsing attempts to construct a *semantic graph* to summarize the meaning of the input text. Since both of them focus on extracting the main information from a sentence, the output information networks and semantic graphs have a lot in common in terms of node and edge semantics. In an example shown in Figure 1, many knowledge elements in the information network can be perfectly matched to certain nodes in the semantic graph with similar semantic meanings. Moreover, these two types of graphs may also be similar with regard to network topology. Specifically, the nodes that are

<sup>1</sup>The programs are publicly available for research purpose at <https://github.com/zhangzx-uiuc/AMR-IE>.

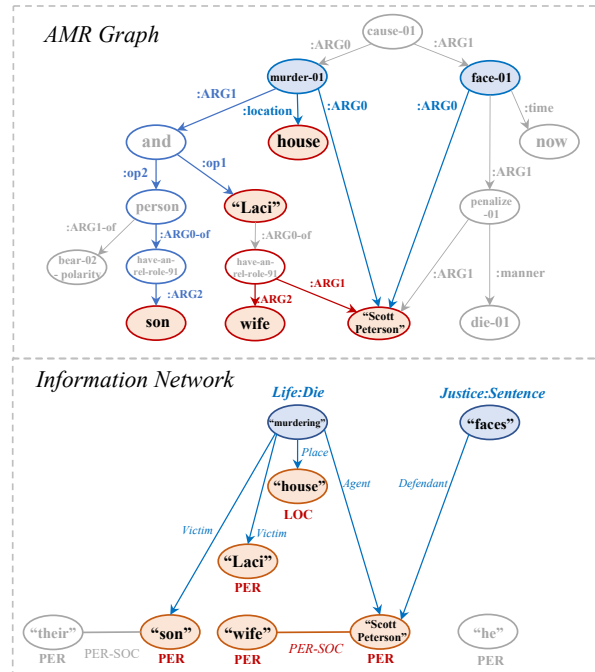


Figure 1: Comparison of the AMR graph generated from pre-trained AMR parser and information network from IE for the same sentence from ACE05: *Scott Peterson now faces death penalty because of murdering his wife Laci and their unborn son at their house.*

neighbors or connected via a few hops in the semantic graph are also likely to be close to each other in the corresponding information network. In Figure 1 we can see that “*Scott Peterson*”, which acts as a shared argument for two event triggers “*murdering*” and “*faces*”, is also directly linked to two main predicates *murder-01* and *face-01* in the semantic graph. From a global perspective, an information network can be approximately considered as a sub-graph of semantic parsing, where the IE nodes are roughly a subset of the nodes in the semantic graph while maintaining similar inter-connections.

To further exploit and make use of such similarities for information extraction, we propose an intuitive and effective framework to utilize information from semantic parsing to jointly extract an in-

formation network composed of entities, relations, event triggers and their arguments. We adopt Abstract Meaning Representation (AMR) (Banarescu et al., 2013) which contains rich semantic structures with fine-grained node and edge types as our input semantic graphs. Compared with previous IE models, our proposed model mainly consists of the following two novel components.

**AMR-Guided Graph Encoding.** The AMR graph topology can directly inform the IE model some global inter-dependencies among knowledge elements, even if they are located far away in the original sentence. Such a property makes it easier for the IE model to capture some non-local long-distance connections for relation and event argument role labeling. We design a **semantic graph aggregator** based on Graph Attention Networks (GAT) (Velickovic et al., 2018) to let the candidate entity and event trigger nodes to aggregate neighborhood information from the semantic graph for passing message among related knowledge elements. The GAT architecture used in our model is specifically designed to allow interactions between node and edge features, making it possible to effectively leverage the rich edge types in AMR.

**AMR-Conditioned Graph Decoding.** A large number of nodes in these two types of graphs share similar meanings, which makes it possible to obtain a meaningful node alignment between information networks and semantic graphs. Such an alignment provides potential opportunities to design a more organized way in the decoding part of a joint IE model. Instead of using sequential decoding as in previous models like OneIE (Lin et al., 2020), where the types of knowledge elements are determined in a left-to-right order according to their positions in the original sentence, we propose a new **hierarchical decoding** method. We use AMR parsing as a *condition* to decide the order of decoding knowledge elements, where the nodes and edges are determined in a tree-like order based on the semantic graph hierarchy.

Experiment results on multiple datasets show that our proposed model significantly outperforms state-of-the-art on all IE subtasks.

## 2 Problem Formulation

We focus on extracting entities, relations, event triggers and their arguments jointly from an input sentence to form an information network. Note that the AMR graphs in our model are not required

to be ground-truth but are generated by pretrained AMR parsers. Therefore, we do not incorporate additional information and our problem settings are identical to typical joint information extraction approaches such as DyGIE++ (Wadden et al., 2019) and OneIE (Lin et al., 2020). Given an input sentence  $S = \{w_1, w_2, \dots, w_N\}$ , we formulate our problem of joint information extraction as follows.

**Entity Extraction** Entity extraction aims to identify word spans as entity mentions and classify them into pre-defined entity types. Given the set of entity types  $E$ , the entity extraction task is to output a collection  $\mathcal{E}$  of entity mentions:

$$\mathcal{E} = \{\varepsilon_i = (a_i, b_i, e_i) \mid a_i \leq b_i, e_i \in E\}$$

where  $a_i, b_i \in \{1, 2, \dots, N\}$  denote the starting and ending indices of the extracted entity mentions, and  $e_i$  represents the entity type in a type set  $E$ . For example, in Figure 1, the entity mention “*Scott Peterson*” is represented as  $(0, 1, PER)$ .

**Relation Extraction** The task of relation extraction is to assign a relation type to every possible ordered pair in the extracted entity mentions. Given the identified entity mentions  $\mathcal{E}$  and pre-defined relation types  $R$ , the set of relations is extracted as

$$\mathcal{R} = \{r_i = (\varepsilon_i, \varepsilon_j, l_{ij}^r) \mid l_{ij}^r \in R, \varepsilon_i, \varepsilon_j \in \mathcal{E}\}$$

where  $\varepsilon_i$  and  $\varepsilon_j$  are entity mentions from  $\mathcal{E}$  and  $i, j \in \{1, 2, \dots, |\mathcal{E}|\}$ . An example relation mention is (“*their*”, “*son*”,  $PER-SOC$ ) in Figure 1.

**Event Extraction** The task of event extraction includes extracting event triggers and their arguments. Event trigger extraction is to identify the words or phrases that most clearly indicate the occurrence of a certain type of event from an event type set  $T$ , which can be formulated as:

$$\mathcal{T} = \{\tau_i = (p_i, q_i, t_i) \mid p_i \leq q_i, t_i \in T\}$$

where  $p_i, q_i \in \{1, 2, \dots, N\}$  denotes the starting and ending indices of the extracted event mentions, and  $t_i$  represents an event type in  $T$ . Given the pre-defined set of event arguments  $A$ , the task of event argument extraction is to assign each trigger and entity pair an argument role label to indicate if an entity mention acts as some certain role of the event, which is formulated as extracting an argument set  $\mathcal{A}$

$$\mathcal{A} = \{\alpha_i = (\tau_i, \varepsilon_j, l_{ij}^a) \mid l_{ij}^a \in A, \tau_i \in \mathcal{T}, \varepsilon_j \in \mathcal{E}\}$$

where  $\tau_i$  and  $\varepsilon_j$  are previously extracted event and entity mentions respectively, and  $l_{ij}^a$  denotes the event argument role label.

**Information Network Construction** All of these extracted knowledge elements form an information network  $G = (V, E)$  (an example is shown in Figure 1). Each node  $v_i \in V$  is an entity mention or event trigger, and each edge  $e_i \in E$  indicates a relation or event argument role. Thus our problem can be formulated as generating an information network  $G$  given an input sentence  $S$ .

### 3 Our Approach

Given an input sentence  $S$ , we first use a pre-trained transformer-based AMR parser (Fernandez Astudillo et al., 2020) to obtain the AMR graph for  $S$ . We then use RoBERTa (Liu et al., 2019) to encode each sentence to identify entity mentions and event triggers as candidate nodes. After that, we map each candidate node to AMR nodes and enforce message passing using a GAT-based semantic graph aggregator to capture global inter-dependency between candidate nodes. All the candidate nodes and their pairwise edges are then passed through task-specific feed-forward neural networks to calculate score vectors. During decoding, we use the hierarchical structure in each AMR graph as a condition to decide the order in beam search and find the best candidate graph with the highest global score.

#### 3.1 AMR Parsing

We employ a transformer based AMR parser (Fernandez Astudillo et al., 2020) pre-trained on AMR 3.0 annotations<sup>2</sup> to generate an AMR graph  $G^a = (V^a, E^a)$  with an alignment between AMR nodes and word spans in an input sentence  $S$ . Each node  $v_i^a = (m_i^a, n_i^a) \in V^a$  represents an AMR concept or predicate, and we use  $m_i^a$  and  $n_i^a$  to denote the starting and ending indices of such a node in the original sentence. For AMR edges, we use  $e_{i,j}^a$  to denote the specific relation type between nodes  $v_i^a$  and  $v_j^a$  in AMR annotations.

**Embeddings for AMR Relation Clusters** To reduce the risk of over-fitting on hundreds of fine-grained AMR edge types, we only consider the edge types that are most relevant to IE tasks, and manually define  $M = 12$  clusters of AMR edge types as shown in Table 1. Note that each  $ARGx$

relation is considered as an individual cluster since each  $ARGx$  indicates a distinct argument role. For each edge type cluster, we randomly initialize a  $d_E$  dimensional embedding and obtain an embedding matrix  $E \in \mathbb{R}^{M \times d_E}$ , which will be optimized during the training process.

Categories	AMR relation types
Spatial	location, destination, path
Temporal	year, time, duration, decade, weekday
Means	instrument, manner, topic, medium
Modifiers	mod, poss
Operators	op-X
Prepositions	prep-X
Core Roles	ARG0, ARG1, ARG2, ARG3, ARG4
Others	Other AMR relation types.

Table 1: Manually defined AMR relation clusters for IE, where each  $ARGx$  is treated as an individual cluster.

#### 3.2 Entity and Event Trigger Identification

We first identify the entity mentions and event triggers as candidate nodes from an input sentence. Similar to (Lin et al., 2020), we adopt feed forward neural networks constrained by conditional random fields (CRFs) to identify the word spans for entity mentions and event triggers.

**Contextual Encoder** Given an input sentence  $S = \{w_1, w_2, \dots, w_N\}$  of length  $N$ , we first calculate the contextual word representation  $x_i$  for each word  $w_i$  using a pre-trained RoBERTa encoder (Liu et al., 2019). If one word is split into multiple pieces by the RoBERTa tokenizer, we take the average of the representation vectors for all word pieces as the final word representation.

*(can do changes here!)*

**CRFs based Sequence Tagging** After obtaining the contextual word representations, we use a feed-forward neural network  $FFN$  to compute a score vector  $\hat{y}_i = FFN(x_i)$  for each word, where each element in  $\hat{y}_i$  represents the score for a certain tag in the tag set<sup>3</sup>. The overall score for a tag path  $\hat{z} = \{\hat{z}_1, \hat{z}_2, \dots, \hat{z}_N\}$  is calculated by

$$s(\hat{z}) = \sum_{i=1}^N \hat{y}_{i, \hat{z}_i} + \sum_{i=1}^{N+1} P_{\hat{z}_{i-1}, \hat{z}_i},$$

where  $\hat{y}_{i, \hat{z}_i}$  is the  $\hat{z}_i$ -th element of the score vector  $\hat{y}_i$ , and  $P_{\hat{z}_{i-1}, \hat{z}_i}$  denotes the transition score from tag  $\hat{z}_{i-1}$  to  $\hat{z}_i$  from an optimizable matrix  $P$ . Similar to (Chiu and Nichols, 2016), the training

<sup>2</sup><https://catalog.ldc.upenn.edu/LDC2020T02>

<sup>3</sup>We use BIO tagging scheme to tag word spans.

$$\log\left(\frac{e^{s(z)}}{\sum_{\hat{z} \in Z} e^{s(\hat{z})}}\right) = (22)$$

objective for node identification is to maximize the log-likelihood  $\mathcal{L}^I$  of the gold tag-path  $z$ .

$$\mathcal{L}^I = s(z) - \log \sum_{\hat{z} \in Z} e^{s(\hat{z})} \quad (1)$$

We use separate CRF-based taggers for entity and event trigger extraction. Note that we do not use the **specific node types predicted by the CRF taggers as the final output classification results for entities and triggers**, but only keep the identified entity and trigger spans. The final types of entities and triggers are jointly decided with relation and argument extraction in the subsequent decoding step. Specifically, we will obtain the collections of entity spans  $\{(a_i, b_i)\}_{i=1}^{|\mathcal{E}|}$  and trigger spans  $\{(p_i, q_i)\}_{i=1}^{|\mathcal{T}|}$  during this step, where  $a_i, b_i, p_i, q_i$  denote the starting and ending indices of the word spans.

### 3.3 Semantic Graph Aggregator

To make the best use of the shared semantic features and topological features from the AMR parsing for the input sentence, we design a semantic graph aggregator, which enables the candidate entity nodes and event nodes to aggregate information from their neighbors based on the AMR topology.

**Initial Node Representation** Each entity node, trigger node or AMR node is initialized with a vector representation  $h_i^0$  by averaging the word embeddings for all the words in their spans. For example, given an entity node  $(a_i, b_i)$ , its representation vector is calculated by

$$h_i^0 = \frac{1}{|b_i - a_i + 1|} \sum_{k=a_i}^{b_i} x_k$$

where  $x_k$  is the word representation from the RoBERTa encoder.

**Node Alignment** We first try to align each identified entity node and trigger node to one of the AMR nodes before conducting message passing. Take an entity node with its span  $(a_i, b_i)$  as an example. Given the set of AMR nodes  $\{(m_i^a, n_i^a)\}_{i=1}^{|V^a|}$ , we consider  $b_i$  as the index of the *head word* of the entity node, and aim to find  $(m_{i^*}^a, n_{i^*}^a)$  that covers  $b_i$  as the matched AMR node for  $(a_i, b_i)$ , that is, such a node satisfies  $m_{i^*}^a \leq b_i \leq n_{i^*}^a$ . If no nodes can be matched to  $(a_i, b_i)$  in this way, we turn to search for the *nearest AMR node*:

$$i^* = \arg \min_k (|b_i - m_k| + |b_i - n_k|),$$

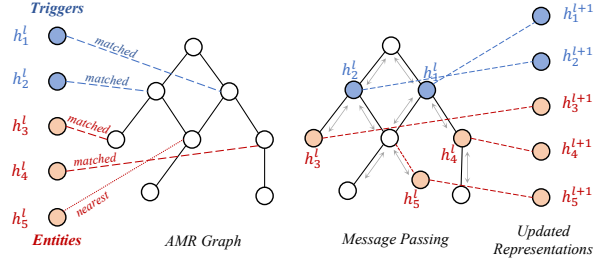


Figure 2: An illustration for node alignment and message passing. For unmatched entity or trigger nodes, such as  $h_5^l$ , we add a new node with feature  $h_5^l$  and link this new node to the *nearest node* of  $h_5^l$ .

where  $(m_{i^*}^a, n_{i^*}^a)$  is the AMR node with the shortest distance to the entity node  $(a_i, b_i)$ . We also conduct alignment for event trigger nodes in the same way.

**Heterogeneous Graph Construction** After obtaining the *matched* or *nearest* AMR node for each identified entity mention and event trigger, we construct a heterogeneous graph with initialized node and edge features as follows. Given an AMR graph  $G^a = (V^a, E^a)$ , we consider the following three cases to initialize feature vectors for each node  $v_i^a$ :

- Node  $v_i^a$  has been *matched* to an entity mention or event trigger. We take the representation vector of the matched node (instead of  $v_i^a$ ) as the initialized feature vector.
- Node  $v_i^a$  is not *matched* to any identified nodes but labeled as the *nearest node* for an entity mention or event trigger, e.g.,  $(a_i, b_i)$ . We add a new node in the AMR topology with the representation vector of  $(a_i, b_i)$ , and link this new node from  $v_i^a$  with an edge type *Others* defined in Table 1.
- Node  $v_i^a$  is neither *matched* nor acted as the *nearest node* to any entities (triggers). We use its own node representation as the initialized feature vector.

For each edge  $e_{i,j}^a$ , we first map it to an AMR relation cluster according to Table 1 and then look up for its representation  $e_{i,j}$  from the embedding matrix  $E$ . We use  $h_i^0$  to represent the initial feature for each node. An illustration for this step is shown in Figure 2.

**Attention Based Message Passing** Inspired from Graph Attention Networks (GATs) (Velickovic et al., 2018), we design an  $L$ -layer attention based message passing mechanism on an AMR graph topology to enable the entity and trigger



nodes to aggregate neighbor information. For the node  $i$  in layer  $l$ , we first calculate the attention score for each neighbor  $j \in \mathcal{N}_i$  based on node features  $\mathbf{h}_i^l, \mathbf{h}_j^l$  and edge features  $\mathbf{e}_{i,j}^l$ .

$$\alpha_{i,j}^l = \frac{\exp(\sigma(f^l[\mathbf{W}\mathbf{h}_i^l : \mathbf{W}_e\mathbf{e}_{i,j} : \mathbf{W}\mathbf{h}_j^l]))}{\sum_{k \in \mathcal{N}_i} \exp(\sigma(f^l[\mathbf{W}\mathbf{h}_i^l : \mathbf{W}_e\mathbf{e}_{i,k} : \mathbf{W}\mathbf{h}_k^l]))}$$

where  $\mathbf{W}, \mathbf{W}_e$  are trainable parameters, and  $f^l$  and  $\sigma(\cdot)$  are a single layer feed-forward neural network and *LeakyReLU* activation function respectively. Then the neighborhood information  $\mathbf{h}^*$  can be calculated by the weighted sum of neighbor features.

$$\mathbf{h}^* = \sum_{j \in \mathcal{N}_i} \alpha_{i,j}^l \mathbf{h}_j^l$$

The updated node feature is calculated by a combination of the original node feature and its neighborhood information, where  $\gamma$  controls the level of message passing between neighbors, and  $\mathbf{W}^*$  denotes a trainable linear transformation parameter.

$$\mathbf{h}^{l+1} = \mathbf{h}^l + \gamma \cdot \mathbf{W}^* \mathbf{h}^* \quad (2)$$

We select the entity and trigger nodes from the graph and take their feature vectors  $\mathbf{h}_i^L$  from the final layer as the representation vectors that have aggregated information from the AMR graph (as Fig. 2 illustrates). We use  $\mathbf{h}_i^e$  and  $\mathbf{h}_i^t$  to denote the features of each entity and trigger respectively.

### 3.4 Model Training and Decoding

In this subsection, we introduce how we jointly decode the output information network given the identified entity and trigger nodes with their aggregated features  $\mathbf{h}_i^e$  and  $\mathbf{h}_i^t$ . We design a hierarchical decoding method that incorporates the AMR hierarchy as a *condition* to decide a more organized order for decoding knowledge elements.

**Maximizing Scores with Global Features** Similar to OneIE (Lin et al., 2020), we use task-specific feed-forward neural networks to map each node or node pair into a score vector. Specifically, we calculate four types of score vectors  $\mathbf{s}_i^e, \mathbf{s}_i^t, \mathbf{s}_{i,j}^r$  and  $\mathbf{s}_{i,j}^a$  for entity, trigger, relation, and argument role extraction tasks respectively, where the dimension of each score vector is identical to the number of classes in each task.

$$\begin{aligned} \mathbf{s}_i^e &= FFN^e(\mathbf{h}_i^e), \quad \mathbf{s}_i^t = FFN^t(\mathbf{h}_i^t), \\ \mathbf{s}_{i,j}^r &= FFN^r([\mathbf{h}_i^e : \mathbf{h}_j^e]), \\ \mathbf{s}_{i,j}^a &= FFN^a([\mathbf{h}_i^t : \mathbf{h}_j^e]). \end{aligned}$$

Therefore, the total score  $c(G)$  is formulated as

$$c(G) = \sum_{i=1}^{|\mathcal{E}|} \mathbf{s}_i^e + \sum_{i=1}^{|\mathcal{T}|} \mathbf{s}_i^t + \sum_{i=1}^{|\mathcal{E}|} \sum_{j=1}^{|\mathcal{E}|} \mathbf{s}_{i,j}^r + \sum_{i=1}^{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{E}|} \mathbf{s}_{i,j}^a.$$

We inherit the approach of using global features in OneIE (Lin et al., 2020) to enforce the model to capture more information on global interactions. The global score  $g(G)$  for an information network  $G$  is defined as the sum of local score  $c(G)$  and the contribution of global features  $\mathbf{f}_G$ .

$$g(G) = c(G) + \mathbf{u} \cdot \mathbf{f}_G \quad (3)$$

where  $\mathbf{u}$  is a trainable parameter. The global feature vector  $\mathbf{f}_G$  is composed of binary values indicating whether the output graph possesses some interdependencies among knowledge elements (e.g., an attacker is likely to be a person being arrested). We use the global feature categories identical to (Lin et al., 2020) during training, and the overall training objective is to maximize the identification log-likelihood, the local score  $s(G)$  while minimizing the gap on the global score between ground-truth  $\hat{G}$  and predicted information network  $G$ .

$$\max \mathcal{L}^I + c(\hat{G}) - (g(\hat{G}) - g(G)).$$

**Hierarchical Ordered Decoding** Given the output score vectors for all nodes and their pairwise edges, the most straightforward way is to output an information network  $G$  with the highest global score  $g(G)$ . Due to the utilization of global features, searching through all possible information networks could incur exponential complexity, thus we take a similar approach based on beam search used in (Lin et al., 2020). Compared with OneIE (Lin et al., 2020), we creatively incorporate the AMR hierarchy to decide a more organized decoding order instead of a simple left-to-right order based on the word positions in the original sentence. Specifically, given the nodes and their alignments with AMR, we sort up these nodes according to the positions of their aligned AMR nodes in a **top-to-down manner**, that is, the aligned AMR node which is nearest to the AMR root node needs to be decoded first. We illustrate the decoding order in Fig. 3 using an example. We use  $\mathcal{U} = \{v_1, v_2, \dots, v_K\}$  to denote the sorted identified trigger and entity nodes, and similar to (Lin et al., 2020), we add these nodes step by step from  $v_1$  to  $v_K$ , and in each step, we obtain all possible subgraphs by enumerating the types of the new

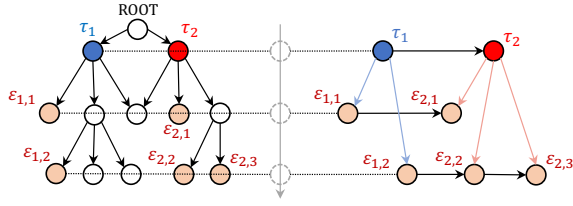


Figure 3: An illustration of ordered decoding, where  $\tau_1$  and  $\tau_2$  are identified triggers while each  $\epsilon_{i,j}$  is identified entity. In this example, the order of beam search decoding is:  $\tau_1, \tau_2, \epsilon_{1,1}, \epsilon_{2,1}, \epsilon_{1,2}, \epsilon_{2,2}, \epsilon_{2,3}$ .

node and pairwise edges with other existing nodes. We only keep the top  $\theta$  subgraphs in each step as candidate graphs to avoid exponential complexity before finally select the graph with the highest global score  $g(G)$  in step  $K$  as the output.

## 4 Experiments

### 4.1 Data

**ACE-2005** Automatic Content Extraction (ACE) 2005 dataset<sup>4</sup> provides fine-grained annotations for entity, relation, and event extraction. We use the same preprocessing and data split as in OneIE (Lin et al., 2020) and DyGIE++ (Wadden et al., 2019) to obtain the ACE05-E corpus with 18,927 sentences. Following (Lin et al., 2020), we keep 7 entity types, 6 relation types, 33 event types, and 22 event argument roles.

**ERE-EN** We also adopt another dataset ERE-EN from the Deep Exploration and Filtering of Test (DEFT) program, which includes more recent news articles and political reviews. We extract 17,108 sentences from datasets LDC2015E29, LDC2015E68, and LDC2015E78. Following (Lin et al., 2020), we keep 7 entity types, 5 relation types, 38 event types, and 20 argument roles.

**GENIA** To further prove that our proposed model is generalizable to other specific domains, we also evaluate our model on biomedical event extraction datasets BioNLP Genia 2011 and 2013 (Kim et al., 2011, 2013). We ignore all of the trigger-trigger links (nested event structures) and merge all repeated event triggers into unified information networks to make them compatible for comparison with previous models. Since the test sets are blind and not available for merging the annotations, we evaluate the model performance on the official development sets instead. Details of dataset statistics are shown in Table 2.

<sup>4</sup><https://catalog.ldc.upenn.edu/LDC2006T06>

Dataset	Split	#Sents	#Ents	#Events	#Rels
ACE05-E	Train	17,172	29,006	4,202	4,664
	Dev	923	2,451	450	560
	Test	832	3,017	403	636
ERE-EN	Train	14,736	39,501	6,208	5,054
	Dev	1,209	3,369	525	408
	Test	1,163	3,295	551	466
Genia'11	Train	9,583	12,058	5,854	513
	Dev	3,499	4,842	1,933	117
Genia'13	Train	2,992	3,794	1,776	46
	Dev	3,341	4,542	1,821	34

Table 2: Dataset statistics.

### 4.2 Experimental Setup

We adopt the most recent joint IE models DyGIE++ (Wadden et al., 2019) and OneIE (Lin et al., 2020) as baselines in our experiments, and use the same evaluation metrics as (Zhang et al., 2019b; Wadden et al., 2019; Lin et al., 2020) to report the F1-Score for each IE subtask.

**Entity:** An extracted entity mention is correct only if both the predicted word span  $(a_i, b_i)$  and entity type  $e_i$  match a reference entity mention.

**Event Trigger:** An event trigger is correctly identified (Trg-I) if the predicted span  $(p_i, q_i)$  matches a reference trigger. It is correctly classified (Trg-C) if the predicted event type  $t_i$  also matches the reference trigger.

**Event Argument:** A predicted event argument  $(\tau_i, \epsilon_j, l_{i,j}^a)$  is correctly identified (Arg-I) if  $(\tau_i, \epsilon_j)$  matches a reference event argument. It is correctly classified (Arg-C) if the type  $l_{i,j}^a$  also matches the reference argument role.

**Relation:** A predicted relation is correct only if its arguments  $\epsilon_i$  and  $\epsilon_j$  both match a reference relation mention.

We train our model with Adam (Kingma and Ba, 2015) on NVIDIA Tesla V100 GPUs for 80 epochs (approximately takes 10 minutes for 1 training epoch) with a learning rate  $1e-5$  for RoBERTa parameters and  $5e-3$  for other parameters. We take the level of message passing  $\gamma$  as 0.001, which is a relatively low level of message passing because we found that too much message passing will result in the loss of own features for the nodes. We use a two-layer semantic graph aggregator and the feature dimensions are 2048 for nodes and 256 for edges. For other hyper-parameters, we keep them strictly identical to (Lin et al., 2020) to enforce fair comparison. Specifically, the FFNs consist of two layers with a dropout rate of 0.4, where the num-

Dataset	ACE05-E						ERE-EN					
Tasks	Ent	Trg-I	Trg-C	Arg-I	Arg-C	Rel	Ent	Trg-I	Trg-C	Arg-I	Arg-C	Rel
DyGIE++	89.7	-	69.7	53.0	48.8	-	-	-	-	-	-	-
OneIE	90.2	77.9	74.7	57.9	55.6	61.8	86.3	66.0	57.1	43.7	42.1	52.8
AMR-IE w/o Enc	90.3	77.9	74.8	58.8	56.6	61.8	86.5	66.2	57.1	44.8	43.0	53.0
AMR-IE w/o Dec	91.9	78.1	74.9	59.0	57.8	62.2	87.8	67.6	60.9	45.6	44.1	54.4
AMR-IE (Ours)	<b>92.1</b>	<b>78.1</b>	<b>75.0</b>	<b>60.9</b>	<b>58.6</b>	<b>62.3</b>	<b>87.9</b>	<b>68.0</b>	<b>61.4</b>	<b>46.4</b>	<b>45.0</b>	<b>55.2</b>

Table 3: Overall test F-scores (%) of joint information extraction. *AMR-IE w/o Enc* and *AMR-IE w/o Dec* are model ablation variants where we only keep the ordered decoding and graph encoding respectively.

bers of hidden units are 150 for entity and relation extraction and 600 for event extraction, and the beam size is set to 10.

### 4.3 Overall Performance

We report the performance of our AMR-IE model and compare it with previous methods in Table 3 and Table 4. In general, our AMR guided method greatly outperforms the baselines on all IE subtasks including entity, event, and relation extraction. The performance improvement is particularly significant on edge classification tasks such as relation extraction and event argument role labeling, because the model can better understand the relations between knowledge elements with the help of external AMR graph structures. To further show the help of each individual part in our model, we introduce two variants of our model for ablation study and show the results in Table 3. In *AMR-IE w/o Enc*, we remove the semantic graph aggregator and only keep the ordered decoding, while in *AMR-IE w/o Dec*, we keep the semantic graph aggregator but use a flat left-to-right decoding order. From the results, we can see that only incorporating the graph encoder is already able to substantially improve the performance on all IE subtasks, because the identified nodes can capture some global interactions through message passing on the AMR topology. Moreover, using an AMR-guided decoding order could further boost the performance especially on the task of event argument extraction.

### 4.4 Influence of Message Passing

We also conduct parameter sensitivity analysis to study the influence of  $\gamma$  defined in Eq. (2), which controls how much information to aggregate from the neighbor nodes in the AMR graph. We change this parameter from  $10^{-5}$  to  $10^1$  and show the performance trends of IE subtasks on ACE-05E dataset in Fig. 4. We can discover that for each subtask, the model performance experiences an in-

Dataset	Model	Ent	Trg-C	Arg-C	Rel
Genia'11	OneIE	81.8	56.9	57.0	63.1
	AMR-IE	<b>82.2</b>	<b>61.5</b>	<b>59.8</b>	<b>65.2</b>
Genia'13	OneIE	71.5	57.3	51.4	39.3
	AMR-IE	<b>78.4</b>	<b>63.8</b>	<b>58.0</b>	<b>42.4</b>

Table 4: Dev set F-scores (%) for joint information extraction on BioNLP Genia 2011 and 2013 datasets.

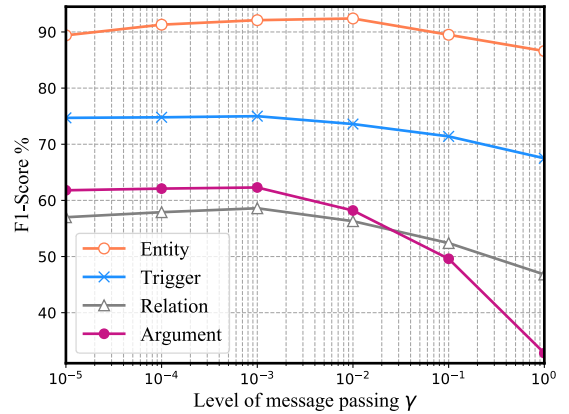


Figure 4: Performance on ACE05-E dataset changes with the level of message passing.

crease as the level of message passing goes stronger. However, when  $\gamma$  continually increases higher than  $10^{-2}$ , the performance of all of the subtasks will undergo a clear decrease. Such a phenomenon follows our intuition since the identified nodes can collect useful information from their AMR neighbors by message passing. However, if the nodes focus too much on their neighborhood information, they will lose some of their own inherent semantic features which results in a performance decrease. In addition, we can also see that compared with entity and trigger extraction tasks, the performance of relation and argument extraction tasks varies more drastically with  $\gamma$ . This is because edge type prediction requires high-quality embeddings for both of the involved nodes, which makes the edge type prediction tasks more sensitive to message passing.

Sentence	AMR Parsing	OneIE outputs	AMR-IE outputs
If the resolution is not passed, <b>Washington</b> would likely want to use the airspace for strikes against Iraq and for <b>airlifting troops</b> to northern <b>Iraq</b> .			
A Pakistani <b>court</b> in central Punjab <b>province</b> has <b>sentenced</b> a Christian <b>man</b> to life imprisonment for a blasphemy <b>conviction</b> , police said Sunday.			
Russian President <b>Vladimir Putin</b> 's <b>summit</b> with the <b>leaders</b> of Germany and France may have been a failure that proves there can be no long-term "peace camp" alliance following the end of war in <b>Iraq</b> .			
Major US insurance group <b>AIG</b> is in the final stage of talks to take over General Electric's Japanese life insurance arm in a deal to <b>create</b> <b>Japan</b> 's sixth largest life <b>insurer</b> , reports said Wednesday.			

Table 5: Examples from ACE05-E test set that illustrates how AMR parsing can improve the performance of joint IE. Note that due to the limitation of space, we only show a subset of each ARM graph that are most relevant for generating the correct IE outputs.

#### 4.5 Qualitative Analysis

In order to further understand how our proposed AMR guided encoding and AMR conditioned decoding method help to improve the performance, we select typical examples from the output of our AMR-IE model for illustration in Table 5.

### 5 Related Work

Some recent efforts have incorporated dependency parsing trees into neural networks for event extraction (Li et al., 2019) and relation extraction (Miwa and Bansal, 2016; Poursan Ben Veyseh et al., 2020). For semantic role labeling (SRL), (Stanovsky and Dagan, 2016) manages to exploit the similarity between SRL and open domain IE by creating a mapping between two tasks. (Huang et al., 2016, 2018) employ AMR as a more concise input format for their IE models, but they decompose each AMR into triples to capture the local contextual information between nodes and edges, while the node information is not disseminated in a global graph topology. (Rao et al., 2017) proposes a sub-graph matching based method to extract biomedical events from AMR graphs, while (Li et al., 2020) uses an additional GCN based encoder for obtaining better word representations.

Besides, graph neural networks are also widely

used for event extraction (Liu et al., 2018; Veyseh et al., 2020; Balali et al., 2020; Zhang et al., 2021) and relation and entity extraction (Zhang et al., 2018; Fu et al., 2019; Guo et al., 2019; Sun et al., 2020). Graph neural networks also demonstrate effectiveness to encode other types of intrinsic structures of a sentence, such as knowledge graph (Zhang et al., 2019a; Huang et al., 2020), document-level relations (Sahu et al., 2019; Lockard et al., 2020; Zeng et al., 2020), and self-constructed graphs (Kim and Lee, 2012; Zhu et al., 2019; Qian et al., 2019; Sahu et al., 2020). However, all these approaches focus on single IE tasks while can not scale to extracting a joint information network with entities, relations, and events.

There are some recent efforts that focus on building joint neural models for performing multiple IE tasks simultaneously, such as joint entity and relation extraction (Li and Ji, 2014; Katiyar and Cardie, 2017; Zheng et al., 2017; Bekoulis et al., 2018; Sun et al., 2019; Luan et al., 2019) and joint event and entity extraction (Yang and Mitchell, 2016). DyGIE++ (Wadden et al., 2019) designs a joint model to extract entities, events, and relations based on span graph propagation, while OneIE (Lin et al., 2020) further makes exploits global features to facilitate the model to capture more global interactions. Compared with the flat



encoder in OneIE, our proposed framework leverages a semantic graph aggregator to incorporate information from fine-grained AMR semantics and enforce global interactions in the encoding phase. In addition, instead of a simple left-to-right sequential decoder, we creatively use the AMR hierarchy to decide the decoding order of knowledge elements. Both the AMR-guided graph encoder and decoder are proven highly effective compared to their flat counterparts.

## 6 Conclusions and Future Work

AMR parsing and IE share the same goal of constructing semantic graphs from unstructured text. IE focuses more on a target ontology, and thus its output can be considered as a subset of AMR graph. In this paper, we present two intuitive and effective ways to leverage guidance from AMR parsing to improve IE, during both encoding and decoding phases. In the future, we plan to integrate AMR graph with entity coreference graph so our IE framework can be extended to document level.

## Acknowledgement

This research is based upon work supported in part by U.S. NSF No. 1741634, U.S. DARPA KAIROS Program No. FA8750-19-2-1004, U.S. DARPA AIDA Program No. FA8750-18-2-0014, Air Force No. FA8650-17-C-7715, LORELEI Program No. HR0011-15-C-0115, the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract No. FA8650-17-C-9116. 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, ODNI, IARPA, 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

Ali Balali, Masoud Asadpour, Ricardo Campos, and Adam Jatowt. 2020. [Joint event extraction along shortest dependency paths using graph convolutional networks](#). *Knowl. Based Syst.*, 210:106492.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. [Abstract Meaning Representation](#)

[for sembanking](#). In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. [Adversarial training for multi-context joint entity and relation extraction](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2830–2836, Brussels, Belgium. Association for Computational Linguistics.

Jason P.C. Chiu and Eric Nichols. 2016. [Named entity recognition with bidirectional LSTM-CNNs](#). *Transactions of the Association for Computational Linguistics*, 4:357–370.

Ramón Fernandez Astudillo, Miguel Ballesteros, Tahira Naseem, Austin Blodgett, and Radu Florian. 2020. [Transition-based parsing with stack-transformers](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1001–1007, Online. Association for Computational Linguistics.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. [GraphRel: Modeling text as relational graphs for joint entity and relation extraction](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418, Florence, Italy. Association for Computational Linguistics.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. [Attention guided graph convolutional networks for relation extraction](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251, Florence, Italy. Association for Computational Linguistics.

Kung-Hsiang Huang, Mu Yang, and Nanyun Peng. 2020. [Biomedical event extraction with hierarchical knowledge graphs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1277–1285, Online. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 258–268, Berlin, Germany. Association for Computational Linguistics.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. [Zero-shot transfer learning for event extraction](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.

- Arzoo Katiyar and Claire Cardie. 2017. [Going out on a limb: Joint extraction of entity mentions and relations without dependency trees](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 917–928, Vancouver, Canada. Association for Computational Linguistics.
- Jin-Dong Kim, Yue Wang, Toshihisa Takagi, and Akinori Yonezawa. 2011. [Overview of Genia event task in BioNLP shared task 2011](#). In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 7–15, Portland, Oregon, USA. Association for Computational Linguistics.
- Jin-Dong Kim, Yue Wang, and Yamamoto Yasunori. 2013. [The Genia event extraction shared task, 2013 edition - overview](#). In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 8–15, Sofia, Bulgaria. Association for Computational Linguistics.
- Seokhwan Kim and Gary Geunbae Lee. 2012. [A graph-based cross-lingual projection approach for weakly supervised relation extraction](#). In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 48–53, Jeju Island, Korea. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Diya Li, Lifu Huang, Heng Ji, and Jiawei Han. 2019. [Biomedical event extraction based on knowledge-driven tree-LSTM](#). 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 1421–1430, Minneapolis, Minnesota. Association for Computational Linguistics.
- Manling Li, Alireza Zareian, Qi Zeng, Spencer Whitehead, Di Lu, Heng Ji, and Shih-Fu Chang. 2020. [Cross-media structured common space for multimedia event extraction](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2557–2568. Association for Computational Linguistics.
- Qi Li and Heng Ji. 2014. [Incremental joint extraction of entity mentions and relations](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.
- Qi Li, Heng Ji, Yu Hong, and Sujian Li. 2014. [Constructing information networks using one single model](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1846–1851, Doha, Qatar. Association for Computational Linguistics.
- Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. [A joint neural model for information extraction with global features](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.
- Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. [Jointly multiple events extraction via attention-based graph information aggregation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). In *ArXiv: abs/1907.11692*.
- Colin Lockard, Prashant Shiralkar, Xin Luna Dong, and Hannaneh Hajishirzi. 2020. [ZeroShotCeres: Zero-shot relation extraction from semi-structured webpages](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8105–8117, Online. Association for Computational Linguistics.
- Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. [A general framework for information extraction using dynamic span graphs](#). 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 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.
- Makoto Miwa and Mohit Bansal. 2016. [End-to-end relation extraction using LSTMs on sequences and tree structures](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.
- Amir Poursan Ben Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020. [Exploiting the syntax-model consistency for neural relation extraction](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8021–8032, Online. Association for Computational Linguistics.
- Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. 2019. [GraphIE: A graph-based framework for information extraction](#). 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 751–761, Minneapolis, Minnesota. Association for Computational Linguistics.

- Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé III. 2017. [Biomedical event extraction using Abstract Meaning Representation](#). In *BioNLP 2017*, pages 126–135, Vancouver, Canada,. Association for Computational Linguistics.
- Sunil Kumar Sahu, Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. [Inter-sentence relation extraction with document-level graph convolutional neural network](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4309–4316, Florence, Italy. Association for Computational Linguistics.
- Sunil Kumar Sahu, Derek Thomas, Billy Chiu, Neha Sengupta, and Mohammady Mahdy. 2020. [Relation extraction with self-determined graph convolutional network](#). In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 2205–2208.
- Gabriel Stanovsky and Ido Dagan. 2016. [Creating a large benchmark for open information extraction](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305, Austin, Texas. Association for Computational Linguistics.
- Changzhi Sun, Yeyun Gong, Yuanbin Wu, Ming Gong, Daxin Jiang, Man Lan, Shiliang Sun, and Nan Duan. 2019. [Joint type inference on entities and relations via graph convolutional networks](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1361–1370, Florence, Italy. Association for Computational Linguistics.
- Kai Sun, Richong Zhang, Yongyi Mao, Samuel Mensah, and Xudong Liu. 2020. [Relation extraction with convolutional network over learnable syntax-transport graph](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI2020, New York, NY, USA, February 7-12, 2020*, pages 8928–8935.
- Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. [Graph attention networks](#). In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.
- Amir Pouran Ben Veyseh, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. [Graph transformer networks with syntactic and semantic structures for event argument extraction](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 3651–3661. Association for Computational Linguistics.
- David Wadden, Ulme Wennberg, Yi Luan, and Hananeh Hajishirzi. 2019. [Entity, relation, and event extraction with contextualized span representations](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.
- Bishan Yang and Tom M. Mitchell. 2016. [Joint extraction of events and entities within a document context](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299, San Diego, California. Association for Computational Linguistics.
- Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. [Double graph based reasoning for document-level relation extraction](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1630–1640, Online. Association for Computational Linguistics.
- Junchi Zhang, Qi He, and Yue Zhang. 2021. [Syntax grounded graph convolutional network for joint entity and event extraction](#). *Neurocomputing*, 422:118–128.
- Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. 2019a. [Long-tail relation extraction via knowledge graph embeddings and graph convolution networks](#). 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 3016–3025, Minneapolis, Minnesota. Association for Computational Linguistics.
- Tongtao Zhang, Heng Ji, and Avirup Sil. 2019b. [Joint entity and event extraction with generative adversarial imitation learning](#). *Data Intelligence*, 1(2):99–120.
- Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. [Graph convolution over pruned dependency trees improves relation extraction](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.
- Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. [Joint extraction of entities and relations based on a novel tagging scheme](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.
- Hao Zhu, Yankai Lin, Zhiyuan Liu, Jie Fu, Tat-Seng Chua, and Maosong Sun. 2019. [Graph neural networks with generated parameters for relation extraction](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1331–1339, Florence, Italy. Association for Computational Linguistics.