Inducing Rich Interaction Structures between Words for Document-level Event Argument Extraction

Amir Pouran Ben Veyseh[1], Franck Dernoncourt[2], Quan Tran[2], Varun Manjunatha[2], Lidan Wang[2], Rajiv Jain[2], Doo Soon Kim[2], Walter Chang[2], and Thien Huu Nguyen[1]

[1] Department of Computer and Information Science,
University of Oregon, Eugene OR, USA
{apouranb,thien}@cs.uoregon.edu
[2] Adobe Research, San Jose CA, USA
{franck.dernoncourt,qtran,vmanjuna,lidwang,rajijain,dkim,wachang}@adobe.com

Abstract. Event Argument Extraction (EAE) is the task of identifying roles of entity mentions/arguments in events evoked by trigger words. Most existing works have focused on sentence-level EAE, leaving document-level EAE (i.e., event triggers and arguments belong to different sentences in documents) an under-studied problem in the literature. This paper introduces a new deep learning model for document-level EAE where document structures/graphs are utilized to represent input documents and aid the representation learning. Our model employs different types of interactions between important context words in documents (i.e., syntax, semantic, and discourse) to enhance document representations. Extensive experiments are conducted to demonstrate the effectiveness of the proposed model, leading to the state-of-the-art performance for document-level EAE.

Keywords: Event Argument Extraction · Document Structures

1 Introduction

Event Extraction (EE) is an important and challenging task in Information Exaction (IE) that aims to identify instances of events (i.e., change of states of real-world entities) in text. To this end, two subtasks should be solved: (1) Event Detection (ED) to recognize event-triggering expressions (verbal predicates or nominalizations, called event triggers/mentions), and (2) Event Argument Extraction (EAE) to identify entity mentions that are involved in events (event participants and spatio-temporal attributes, collectively known as event arguments). This work focuses on EAE, a relatively less-explored task for EE (compared to ED). Technically speaking, our EAE task takes as inputs an event trigger and an argument candidate (entity mention), seeking to predict the role that the argument candidate plays in the event mention associated with the trigger. A well performing EAE system will benefit various downstream applications such as Knowledge Base Construction and Question Answering.

Most of the recent work on EAE employs deep learning models to achieve state-of-the-art performance [17]. Unfortunately, these models are often restricted to sentence-level EAE where event triggers and arguments appear in the same sentence. In real world scenarios, arguments of an event might have been presented in sentences other than the sentence that hosts the event trigger in the input document. For instance, in the EE dataset of the DARPA AIDA program (phase 1)[3], 38% of arguments has been shown to be outside the sentences containing the corresponding triggers, i.e., in the document-level context [3]. As such, it is of paramount importance to develop models that can extract arguments of event mentions over the entire documents to provide a more complete view of information for events in documents.

A major challenge in document-level EAE involves long document context that hinders the ability of models to effectively identify important context words (among long word sequences) and link them to event triggers and arguments for role prediction. Recently, a promising approach to address this document context modeling issue has been explored for other related tasks in IE [15, 14, 10] where document structures (i.e., direct interactions between parts of documents) are employed to facilitate the connections and reasoning between important context words for a prediction problem.

Thus, one simple solution towards utilizing document structures for EAE is to exert one of the existing document-level models that has been designed for other related tasks. However, in this work, we show that such prior models have critical constraints that should be addressed to better serve EAE. As such, the existing document-level models have only exploited some (typically one) specific types of information/heuristics to form the edges in document structures, thus failing to leverage a diversity of useful information to enrich document structures in EAE. This is unfortunate as multiple information sources are often required simultaneously to capture necessary interaction information between nodes/words and improve the coverage/performance for EAE models. For instance, consider the following document: "*The foundation said that immediately following the Haitian earthquake, the Embassy of Algeria provided an unsolicited lump-sum fund to the foundation's relief plan. This was a one-time, specific donation to help Haiti and it had **donated** twice to the Clinton Foundation before.*". In this two-sentence document, an EAE system needs to recognize the entity mention "*Embassy of Algeria*" as an argument (of role *Giver*) for the event mention associated with the trigger word "*donated*". To perform this reasoning, the models can utilize the coreference link between "*Embassy of Algeria*" and the pronoun "*it*" (i.e., discourse information) that can be directly connected with the trigger word "*donated*" via an edge in the syntactic dependency tree of the second sentence. Alternatively, if the coreference link cannot be obtained (e.g., due to errors in the coreference resolution systems), EAE models can rely on the close semantic similarity between "*donated*" and "*provided an unsolicited lump-sum fund*" that can be further linked to "*Embassy of Algeria*" via a dependency edge in the first sentence. As such, document-level models might need to jointly capture

[3] https://tac.nist.gov/2019/SM-KBP/data.html

the information from syntax, semantic, and discourse structures to sufficiently encode necessary interactions between words for EAE.

Motivated by this intuition, we propose to combine different information sources to generate effective document structures for our EAE problem, focusing on the knowledge from syntax (i.e., dependency trees), discourse (i.e., coreference links), and semantic similarity. Importantly, for the semantic similarity, in addition to using contextualized representation vectors to compute interaction scores between words as in prior work [10], we propose to further leverage external knowledge bases to enrich document structures for EAE. As such, we link the words in the documents to the entries in some external knowledge bases and exploit the entry similarity in such knowledge bases to obtain word similarity scores for the structures. To our knowledge, this is the first work to employ external knowledge bases to compute document structures for an IE task in the literature.

Given various document structures, how can we effectively combine these structures for EAE? Our main principle for this goal is motivated from the running example where the role reasoning process for the event trigger and argument candidate involves a sequence of interactions with multiple other words, possibly using different types of information at each interaction step, e.g., syntax, discourse or semantic information (called heterogeneous interaction types). To this end, we propose to employ Graph Transformer Networks (GTN) [18] to facilitate the implementation of this multi-hop heterogeneous reasoning idea. More specifically, GTNs fulfill the multi-hop heterogeneous reasoning by multiplying weighted sums of different initial document structures to generate rich combined structures. Finally, the resulting combined structures will be used to learn representation vectors for EAE based on graph convolutional networks (GCN). To our knowledge, this is also the first work that introduces GTN and GCN for document structure computation and representation learning in document-level EAE.

We evaluate the proposed model on two benchmark datasets; one for document-level EAE [3] and one for the closely related task of implicit semantic role labeling. Our experiments demonstrate the effectiveness of the proposed model, establishing new state-of-the-art results on both benchmark datasets.

2 Model

We formulate document-level EAE as a multi-class classification problem. The input to the models is a document $D = w_1, w_2, \dots, w_N$ which consists of multiple sentences, i.e., S_i's. To be comparable with previous work [3], we also use a golden event trigger, i.e., the t-th word of D (w_t), and an argument candidate, i.e., the a-th word of D (w_a), as the inputs (w_t and w_a can occur in different sentences). The goal of EAE is to predict the role of the argument candidate w_a in the event evoked by w_t. Here, the role might be $None$, indicating that w_a is not a participant in the event mention w_t. Our model for EAE involve three major components: (i) Document Encoder to transform the words in D into high dimensional vectors, (ii) Structure Generation to generate initial document structures for EAE, and (iii) Structure Combination to combine the structures and learn representation vectors for role prediction. We provide details for these components below.

2.1 Document Encoder

In the first step, we transform each word $w_i \in D$ into a representation vector x_i that is the concatenation of the following two vectors:

(i) The pre-trained word embedding of w_i: Here, we consider both non-contextualized embeddings, i.e., GloVe and contextualized embeddings, i.e., BERT in the experiments. In particular, for BERT, as w_i might be split into multiple word-pieces, we use the average of the hidden vectors for the word-pieces of w_i in the last layer as the word embedding vector for w_i. Following [3], we employ the BERT$_{base}$ version that divides D into segments of 512 word-pieces to be encoded separately. In our experiments, we fix the parameters of the BERT$_{base}$.

(ii) The position embeddings of w_i: These vectors are obtained by looking up the relative distances between w_i and the trigger and argument words (i.e., $i - t$ and $i - a$ respectively) in a position embedding table. This table is initialized randomly and updated in the training process. Position embedding vectors are important as they notify the model about the positions of the trigger and argument words.

Given the vector sequence $X = x_1, x_2, \ldots, x_N$ to represent the words in D, we further send it to a bidirectional long short-term memory network (LSTM) to generate a more abstract vector sequence $H = h_1, h_2, \ldots, h_N$. Here, h_i is the hidden vector for w_i that is obtained by concatenating the corresponding forward and backward hidden vectors from the bidirectional LSTM. We will use the hidden vectors in H as the inputs for the next computation. Note that we do not include the sentence boundary information of D into the hidden vectors H so far as it will be addressed in our document structures later.

2.2 Structure Generation

The goal of this section is to generate initial document structures that will be combined to learn representation vectors for document-level EAE in the next step. Formally, a document structure in our work involves an interaction graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ between the words in D, i.e., $\mathcal{N} = \{w_i | w_i \in D\}$. As such, the document structure $\mathcal{G}$ can be represented via a real-valued adjacency matrix $A = \{a_{ij}\}_{i,j=1..N}$ where the value/score a_{ij} reflects the importance (or the level of interaction) of w_j for the representation computation of w_i for EAE. As presented in the introduction, we simultaneously consider three types of information to form the edges $\mathcal{E}$ (or compute the interaction scores a_{ij}) in this work, including syntax, semantics, and discourse. We describe initial document structures based on these information types in the following.

Syntax-based Structures: The motivation for this type of document structures is based on sentence-level EAE where dependency parsing trees of input sentences have been used to reveal important context, i.e., via shortest dependency paths to connect event triggers and arguments, and guide the interaction modeling between words for argument role prediction. As such, we expect dependency trees for sentences in D can also be exploited to provide useful information for document structures for EAE. In particular, we propose to leverage dependency relations/connections between pairs of words in D to compute the interaction scores a_{ij}^{dep} in the syntax-based document structure $A^{dep} = \{a_{ij}^{dep}\}_{i,j=1..N}$ for D. Here, two words are more important to each other for representation learning if they are connected in dependency tress. To this end, we first obtain the dependency tree T_i for each sentence S_i in D using an off-the-shelf dependency parser[4]. Afterward, to connect the dependency trees T_i for the sentences, following [5], we create a link between the root node of a tree T_i for S_i with the root node of the tree T_{i+1} for the subsequent sentence S_{i+1}. The resulting graph with linked trees T_i is denoted by T^D. In the next step, motivated by shortest dependency paths in sentence-level EAE, we retrieve the shortest path P^D between the nodes for w_t and w_a in T^D. Finally, we compute the interaction score a_{ij}^{dep} by setting it to 1 if (w_i, w_j) or (w_j, w_i) is an edge in P^D, and 0 otherwise.

Semantic-based Structures: These document structures aim to evaluate the interaction scores in the structures based on the semantic similarity between words (i.e., two words are more important for the representation learning of each other if they are more semantically related). As such, we consider two complementary approaches to capture the semantics of the words in D for semantic-based structure generation, i.e., contextual semantics and knowledge-based semantics.

First, in contextual semantics, we seek to reveal the semantic of a word via the context in which it appears. This suggests the use of the contextualized representation vectors $h_i \in H$ (obtained from the LSTM model) to capture contextual semantics for the words $w_i \in D$ and produce the contextual semantic-based document structure $A^{context} = \{a_{ij}^{context}\}_{i,j=1..N}$ for D. Accordingly, to compute the semantic-based interaction score $a_{ij}^{context}$ for w_i and w_j, we employ the normalized similarity score between their contextualized representation vectors:

$$\begin{aligned} k_i &= U_k h_i, q_i = U_q h_i \\ a_{ij}^{context} &= \exp(k_i q_j) / \sum_{v=1..N} \exp(k_i q_v) \end{aligned} \tag{1}$$

where U_k and U_q are trainable weight matrices, and the biases are omitted in this work for brevity.

Second, in knowledge-based semantics, our goal is to employ the external knowledge of the words from knowledge bases to capture their semantics. We expect that such external knowledge can provide a complementary source of information for the contextual semantics of the words (i.e., external knowledge vs internal context), thereby enriching the document structures for D. To this end, we propose to utilize WordNet, a rich knowledge base for word meanings, to obtain external knowledge for the words in D. Essentially, WordNet involves a network that connects word meanings (i.e., synsets) according to various semantic relations (e.g., synonyms, hyponyms). Each node/synset in WordNet is associated with a textual glossary to provide expert definition about the corresponding meaning.

[4] We use the Stanford Core NLP Toolkit to parse the sentences in this work.

Our first step to generate knowledge-based document structures for D is to map each word $w_i \in D$ to a synset node M_i in WordNet that can be done with a Word Sense Disambiguation (WSD) tool. In this work, we use WordNet 3.0 and the state-of-the-art BERT-based WSD tool in [1] to perform such word-synset mapping. Afterward, to determine knowledge-based interaction scores between two words w_i and w_j in D, we can leverage the similarity scores between the two linked synset nodes M_i and M_j in WordNet. As such, to leverage the rich information embedded in the synset nodes M_i, we introduce two versions of knowledge-based document structures for D based on the glossaries of the synset nodes and the hierarchy structure in WordNet:

(1) The glossary-based structure $A^{gloss} = \{a_{ij}^{gloss}\}_{i,j=1..N}$: Here, for each word $w_i \in D$, we first retrieve the glossary GM_i from the corresponding linked node M_i in WordNet (GM_i can be seen as a sequence of words). A representation vector VM_i is then computed to capture the semantic information in GM_i, by applying the max-pooling operation over the pre-trained GloVe embeddings of the words in GM_i. Finally, the glossary-based interaction score a_{ij}^{gloss} for w_i and w_j is estimated via the similarity between the glossary representations VM_i and VM_j (with the consine similarity): $a_{ij}^{gloss} = cosine(VM_i, VM_j)$.

(2) The WordNet hierarchy-based structure $A^{struct} = \{a_{ij}^{struct}\}_{i,j=1..N}$: The interaction score a_{ij}^{struct} for w_i and w_j in this case relies on the structure-based similarity of the linked synset nodes M_i and M_j in WordNet. Accordingly, we employ the Lin similarity measure for the synset nodes in WordNet for this purpose: $a_{ij}^{struct} = \frac{2*\mathrm{IC}(\mathrm{LCS}(M_i,M_j))}{\mathrm{IC}(M_i)+\mathrm{IC}(M_j)}$ where IC and LCS represent the information content of the synset nodes and the least common subsumer of the two synsets in the WordNet hierarchy (most specific ancestor node), respectively.

Discourse-based Structures: Besides the typical lengths of the input texts, a key difference between document-level and sentence-level EAE involves the presence of multiple sentences in document-level EAE where discourse information (i.e., where the sentences span and how they relate to each other) is helpful to understand the input documents. The goal of this part is to leverage such discourse structures to provide complementary information for the syntax- and semantic-based document structures for EAE. To this end, we propose to exploit two following types of discourse information to generate discourse-based document structures for EAE: (1) the sentence boundary-based document structure $A^{sent} = \{a_{ij}^{sent}\}_{i,j=1..N}$: This document structure concerns the same sentence information of the words in D. The intuition is that two words in the same sentence would involve more useful information for the representation computation of each other than those in different sentences. To implement this intuition, we compute A^{sent} by setting the sentence boundary-based score a_{ij}^{sent} to 1 if w_i and w_j appear in the same sentence in D and 0 otherwise; and (2) the coreference-based document structure $A^{coref} = \{a_{ij}^{coref}\}_{i,j=1..N}$: Instead of considering within-sentence information as in A^{sent}, this document structure exploits relations/connections between sentences (cross-sentence information) in D. To this end, we consider two sentences in D as being related if they contain entity mentions that refer to the same entity in D (coreference information)[5]. Given such a relation between sentences, we consider two words in D to be more relevant to each other if they appear in related sentences. To this end, for the coreference-based structure, a_{ij}^{coref} is set to 1 if w_i and w_j appear in different, but related sentences; and 0 otherwise.

2.3 Structure Combination

Up to this point, we have generated six different document structures for D (i.e., $\mathcal{A} = [A^{dep}, A^{context}, A^{gloss}, A^{struct}, A^{sent}, A^{coref}]$). As these document structures are based on complementary types of information (called structure types), this section aims to combine them to generate richer document structures for EAE. Our key intuition to achieve such a combination is to note that each importance score a_{ij}^v in one of the structures A_{ij}^v ($v \in V = \{dep, context, gloss, struct, sent, coref\}$) only considers the direct interaction between the two involving words w_i and w_j (i.e., not including any other words) according to one specific information type v. As motivated in the introduction, we expect each importance score in the combined structures to further condition on interactions with other important context words in D (i.e., in addition to the two involving words) where each interaction between a pair of words can flexibly use any of the six structure types (multi-hop and heterogeneous-type reasoning). To this end, we propose to use Graph Transformer Networks (GTN)[18] to enable such a multi-hop and heterogeneous-type reasoning in the structure combination for EAE.

In particular, to enable multi-hop reasoning paths at different lengths, we first add the identity matrix I (of size $N \times N$) into the set of initial document structures $\mathcal{A} = [A^{dep}, A^{context}, A^{gloss}, A^{struct}, A^{sent}, A^{coref}, I] = [\mathcal{A}_1, \ldots, \mathcal{A}_7]$. The GTN model is then organized into C channels for structure combination, where the k-th channel contains M intermediate document structures $Q_1^k, Q_2^k, \ldots, Q_M^k$ of size $N \times N$. As such, each intermediate structure Q_i^k is computed by a linear combination of the initial structures in $\mathcal{A}$ using learnable weights α_{ij}^k: $Q_i^k = \sum_{j=1..7} \alpha_{ij}^k \mathcal{A}_j$. Here, due to the linear combination, the interaction scores in Q_i^k are able to reason with any of the six initial structure types in V (although such scores still consider the direct interactions of the involving words only). Afterward, the intermediate structures $Q_1^k, Q_2^k, \ldots, Q_M^k$ in each channel k are multiplied to generate the final document structure $Q^k = Q_1^k \times Q_2^k \times \ldots \times Q_M^k$ of size $N \times N$ (for the k-the channel). As shown in [18], the matrix multiplication enables the importance score between a pair of words w_i and w_j in Q^k to condition on the multi-hop interactions/reasoning between the two words and other words in D (up to $M-1$ hops due to the inclusion of I in $\mathcal{A}$). The interactions involved in one importance score in Q^k can also realize any of the initial structure types in V (heterogeneous reasoning) due to the flexibility of the intermediate structure Q_i^k.

Given the rich document structures $Q^1, Q^2, \ldots, Q^C$ from the C channels, GTN then feed them into C graph convolutional networks (GCN) [6] to induce document structure-enriched representation vectors for argument role prediction

[5] We use the Stanford Core NLP Toolkit to determine the coreference of entity mentions.

in EAE (one GCN for each $Q^k = \{Q^k_{ij}\}_{i,j=1..N}$). As such, each of these GCN models involve G layers that produces the hidden vectors $\bar{h}^{k,t}_1, \ldots, \bar{h}^{k,t}_N$ at the t-th layer of the k-th GCN model for the words in D ($1 \leq k \leq C$, $1 \leq t < G$):

$$\bar{h}^{k,t}_i = ReLU(U^{k,t} \sum_{j=1..N} \frac{Q^k_{ij} \bar{h}^{k,t-1}_j}{\sum_{u=1..N} Q^k_{iu}}) \quad (2)$$

where $U^{k,t}$ is the weight matrix for the t-th layer of the k-th GCN model and the input vectors $\bar{h}^{k,0}_i$ for the GCN models are obtained from the contextualized representation vectors H (i.e., $\bar{h}^{k,0}_i = h_i$ for all $1 \leq k \leq C$, $1 \leq i \leq N$).

In the next step, the hidden vectors in the last layers of all the GCN models (at the G-th layers) for w_i (i.e., $\bar{h}^{1,G}_i, \bar{h}^{2,G}_i, \ldots, \bar{h}^{C,G}_i$) are concatenated form the final representation vector h'_i for w_i in the proposed GTN model: $h'_i = [\bar{h}^{1,G}_i, \bar{h}^{2,G}_i, \ldots, \bar{h}^{C,G}_i]$.

Finally, to predict the argument role for w_a and w_t in D, we assemble a representation vector R based on the hidden vectors for w_a and w_t from the GTN model via: $R = [h'_a, h'_t, MaxPool(h'_1, h'_2, \ldots, h'_N)]$. This vector is then sent to a two-layer feed-forward network with softmax in the end to produce a probability distribution $P(.|D, a, t)$ over the possible argument roles. We then optimize the negative log-likelihood L_{pred} to train the model: $\mathcal{L} = -\log P(y|D, a, t)$ where y is the golden argument role for the input example. We call the proposed model the Multi-hop Reasoning for Event Argument extractor with heterogeneous Document structure types (MREAD) for convenience.

3 Experiments

Dataset & Parameters: We evaluate the document-level EAE models in this work on RAMS, a recent dataset introduced in [3] for document-level EAE. RAMS contains 9,124 annotated event mentions across 139 types for 65 argument roles, serving as the largest available dataset for document-level EAE. We use the official train/dev/test split and evaluation script for RAMS provided by [3] to achieve a fair comparison. In addition, we evaluate the models on the BNB dataset [4] for implicit semantic role labelling (iSRL), a closely related task to document-level EAE where the models need to predict roles of argument candidates for a given predicate (arguments and predicates can appear in different sentences in iSRL). In our experiments, we use the version of BNB prepared by [3] (with the same data split and pre-processing script) for a fair comparison. This dataset annotates 2,603 argument mentions for a total of 12 argument roles (for 1,247 predicates/triggers). We use the development set of the RAMS dataset to fine-tune the hyper-parameters of the proposed model MREAD.
Results: We compare our model MREAD with two categories of baselines on RAMS:

(1) Structure-free: These baselines do not exploit document structures for EAE. In particular, we compare MREAD with the RAMS$_{model}$ model in [3] and the **Head-based** model in [19]. Here, RAMS$_{model}$ currently has the state-of-the-art (SOTA) performance for document-level EAE on RAMS.

Model	Standard Decoding			Type Constrained		
	P	R	F1	P	R	F1
RAMS	62.8	74.9	68.3	78.1	69.2	73.3
Head-based	71.5	66.2	68.8	81.1	66.2	73.0
iDepNN	65.8	68.0	66.9	77.1	67.7	72.1
EoG	71.0	71.7	71.4	82.4	69.2	75.2
GCNN	72.2	72.8	72.5	85.1	69.4	76.5
LSR	72.6	73.6	73.1	83.9	71.4	77.2
MREAD (ours)	75.7	75.3	**75.5**	88.2	72.1	**79.3**

Table 1. Performance on the RAMS test set using BERT.

(2) Structure-based: These baselines employ some forms of document structures (mostly based on syntax and semantic information) to learn representation vectors for input documents. Note that as none of the prior work has explored document structure-based models for document-level EAE, we compare MREAD with the existing document structure-based models for a related task of document-level relation extraction (DRE) in IE. As such, the following SOTA models for DRE are considered in this category: (i) **iDepNN** [5]; (ii) **GCNN** [14]: This baseline generates document structures based on both syntax and discourse information (e.g., dependency trees, coreference links). Note that although GCNN also considers more than one source of information for document structures as we do, it fails to exploit semantic-based document structures (for both contextual and knowledge-based semantics) and lacks effective mechanisms for structure combination (i.e., not using GTN); (iii) **LSR** [10]; and (iv) **EoG** [2].

In addition to the standard decoding (i.e., using `argmax` with $P(.|D, a, t)$ to obtain the predicted roles), following [3], we also consider the decoding setting where the models' predictions are constrained to the permissible roles for the event type e evoked by the trigger w_t. Tables 1 and 2 show the the models' performance on the RAMS test set using BERT and GloVe embeddings, respectively. There are several observations from these tables. First, the proposed model MREAD significantly outperforms all the baselines in both the standard and type constrained decoding regardless of the used embeddings (BERT or GloVe). This consistent performance improvement is significant with $p < 0.01$ and clearly demonstrates the effectiveness of MREAD for document-level EAE. Second, except for iDepNN, all the structure-based models significantly outperform the structure-free baselines. This finding is significant especially considering that the structure-based models are not originally designed for document-level EAE, thereby clearly showing the benefits of document structures for document-level EAE. Finally, compared to GCNN and EoG that also consider multiple sources of information as our model, MREAD achieves substantially better performance, suggesting the advantages of contextual and knowledge-based structures along with multi-hop heterogeneous reasoning in our EAE problem.

Finally, we evaluate the performance of MREAD on the BNB dataset for iSRL. As we use the data version prepared by [3] that involves a different train/dev/test split from the original BNB dataset in [4], we directly use the RAMS$_{model}$ model in [3] as our baseline for a fair comparison. In addition, we report the performance of the structure-based baselines (iDepNN, GCNN, LSR, and EoG) for a complete

Model	Standard Decoding			Type Constrained		
	P	R	F1	P	R	F1
RAMS	66.3	69.8	68.0	77.4	68.8	72.9
Head-based	70.2	63.4	66.6	74.6	65.3	69.6
iDepNN	65.7	65.4	65.5	75.7	63.2	68.9
EoG	69.2	69.0	69.1	81.3	68.0	74.1
GCNN	71.1	70.9	71.0	83.7	68.1	75.1
LSR	72.5	72.0	72.2	82.9	70.3	76.1
MREAD (ours)	73.6	73.5	**73.5**	86.7	71.0	**78.1**

Table 2. Performance on the RAMS test set using GloVe.

view. Table 3 shows the performance of the models on the BNB test dataset (using BERT embeddings). As can be seen, MREAD is also better than all the baseline models substantially and significantly ($p < 0.01$), further confirming the benefits of our proposed model in this work.

Model	P	R	F1
RAMS	-	-	76.6
iDepNN	80.0	75.1	77.5
EoG	78.5	74.4	76.4
GCNN	81.0	73.9	77.3
LSR	80.3	74.1	77.1
MREAD (ours)	82.9	75.0	**78.8**

Table 3. Performance on the BNB test set for iSRL.

Ablation Study: Our proposed model combines different types of document structures (i.e., six types in $\mathcal{A}$) using GTN to enable multi-hop and heterogeneous reasoning for document-level EAE. This section studies the contribution of the proposed document structures and structure combination in MREAD by evaluating the performance of the ablated versions of the model on the development set of the RAMS dataset. In particular, the following ablated models are examined: (i) **MREAD-**A^v: In this group of ablated models, we eliminate each of the document structures in $\mathcal{A}$ from MREAD and evaluate the performance of the model with the remaining structures (e.g., **MREAD-**A^{dep}, **MREAD-**A^{sent}, etc.), (ii) **MREAD-GTN**: In this ablated model, the GTN architecture is excluded from MREAD, so the GCN models are directly and separately applied to each document structure in $\mathcal{A}$. (iii) **MREAD-Multi-hop**: This ablated model is to show the effectiveness of multi-hop heterogeneous reasoning/interaction for EAE. As such, this model avoids the multiplication of the intermediate structures Q_i^k in each channel of GTN, and directly runs the GCN models over the intermediate document structures Q_i^k (i.e., the final structures Q^k are not produced).

Table 4 presents the performance of the models on the RAMS development set. As can be seen from the table, the removal of any document structures in $\mathcal{A}$ would significantly hurt the performance of MREAD, thus confirming the effectiveness of the introduced document structures for EAE. Also, the significantly better performance of MREAD over MREAD-Multi-hop suggests that the multiplication of the intermediate structures in the channels of GTN is helpful to generate richer structures for EAE (i.e., by enabling multi-hop heterogeneous reasoning/interactions of words).

Model	P	R	F1
MREAD	75.5	76.5	**76.0**
MREAD-A^{def}	73.5	74.9	74.2
MREAD-$A^{context}$	72.7	73.5	73.1
MREAD-A^{gloss}	74.6	73.4	74.0
MREAD-A^{struct}	74.1	74.3	74.2
MREAD-A^{sent}	72.8	73.2	73.0
MREAD-A^{coref}	73.2	74.9	74.1
MREAD-GTN	72.1	73.7	72.9
MREAD-Multi-Hop	73.2	74.6	73.9

Table 4. Performance of the models on the RAMS development set using BERT embeddings and standard decoding.

4 Related Work

Most of prior work on EE has focused on sentence-level EAE [9, 11, 12, 8, 7, 13]. Recently, some work has considered document-level EAE, featuring [3] as the most related work to our problem. However, the model proposed by [3] (i.e., RAMS_{model}) does not consider document structures to improve the performance for document-level EAE as we do in this work. Our work is also related to the recent document structure-based models for other NLP tasks [2, 15, 16]. However, compared to our proposed model, these prior works on document structures fail to exploit external knowledge to generate the structures and do not involve mechanisms to combine multiple structures for multi-hop heterogeneous reasoning.

5 Conclusion

This work presents a novel deep learning model for document-level EAE. To facilitate the interaction of important context words in the documents for EAE, our model leverages multiple sources of information, including the novel employment of external knowledge bases, to generate document structures to provide effective knowledge for representation learning in EAE. Also, for the first time in EAE, graph transformer networks are employed to produce richer document structures. The experiments confirm the benefits of the proposed model, yielding to SOTA performance on benchamrk datasets.

Acknowledgments

This research has been supported by Vingroup Innovation Foundation (VINIF) in project VINIF.2019.DA18 and Army Research Office (ARO) grant W911NF-17-S-0002. This research is also based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the Better Extraction from Text Towards Enhanced Retrieval (BETTER) Program. The views contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO, ODNI, IARPA, the Department of Defense, or the U.S. Government.

References

1. Blevins, T., Zettlemoyer, L.: Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In: ACL (2020)
2. Christopoulou, F., Miwa, M., Ananiadou, S.: Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In: EMNLP (2019)
3. Ebner, S., Xia, P., Culkin, R., Rawlins, K., Van Durme, B.: Multi-sentence argument linking. In: ACL (2020)
4. Gerber, M., Chai, J.Y.: Semantic role labeling of implicit arguments for nominal predicates. In: Computational Linguistics (2012)
5. Gupta, P., Rajaram, S., Schütze, H., Runkler, T.: Neural relation extraction within and across sentence boundaries. In: AAAI (2019)
6. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
7. Lai, V.D., Nguyen, T.N., Nguyen, T.H.: Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In: EMNLP (2020)
8. Le, D.M., Nguyen, T.H.: Fine-grained event trigger detection. In: EACL (2021)
9. Li, Q., Ji, H., Huang, L.: Joint event extraction via structured prediction with global features. In: ACL (2013)
10. Nan, G., Guo, Z., Sekulic, I., Lu, W.: Reasoning with latent structure refinement for document-level relation extraction. In: ACL (2020)
11. Nguyen, T.H., Cho, K., Grishman, R.: Joint event extraction via recurrent neural networks. In: NAACL-HLT (2016)
12. Nguyen, T.M., Nguyen, T.H.: One for all: Neural joint modeling of entities and events. In: AAAI (2019)
13. Pouran Ben Veyseh, A., Nguyen, T.N., Nguyen, T.H.: Graph transformer networks with syntactic and semantic structures for event argument extraction. In: EMNLP Findings (2020)
14. Sahu, S.K., Christopoulou, F., Miwa, M., Ananiadou, S.: Inter-sentence relation extraction with document-level graph convolutional neural network. In: ACL (2019)
15. Thayaparan, M., Valentino, M., Schlegel, V., Freitas, A.: Identifying supporting facts for multi-hop question answering with document graph networks. In: The thirteenth Workshop on Graph-Based Methods for Natural Language Processing at EMNLP (2019)
16. Tran, H.M., Nguyen, M.T., Nguyen, T.H.: The dots have their values: Exploiting the node-edge connections in graph-based neural models for document-level relation extraction. In: EMNLP Findings (2020)
17. Wang, X., Wang, Z., Han, X., Liu, Z., Li, J., Li, P., Sun, M., Zhou, J., Ren, X.: Hmeae: Hierarchical modular event argument extraction. In: EMNLP-IJCNLP (2019)
18. Yun, S., Jeong, M., Kim, R., Kang, J., Kim, H.J.: Graph transformer networks. In: NeurIPS (2019)
19. Zhang, Z., Kong, X., Liu, Z., Ma, X., Hovy, E.: A two-step approach for implicit event argument detection. In: ACL (2020)