Document-Level Event Argument Extraction via Optimal Transport

Amir Pouran Ben Veyseh[1], **Minh Van Nguyen**[1],
Franck Dernoncourt[2], **Bonan Min**[3], **Thien Huu Nguyen**[1]
[1]Department of Computer and Information Science,
University of Oregon, Eugene, OR, USA
[2]Adobe Research, San Jose, CA, USA
[3] Raytheon BBN Technologies, USA
{apouranb,minhnv,thien}@cs.uoregon.edu
franck.dernoncourt@adobe.com, bonan.min@raytheon.com

Abstract

Event Argument Extraction (EAE) is one of the sub-tasks of event extraction, aiming to recognize the role of each entity mention toward a specific event trigger. Despite the success of prior works in sentence-level EAE, the document-level setting is less explored. In particular, whereas syntactic structures of sentences have been shown to be effective for sentence-level EAE, prior document-level EAE models totally ignore syntactic structures for documents. Hence, in this work, we study the importance of syntactic structures in document-level EAE. Specifically, we propose to employ Optimal Transport (OT) to induce structures of documents based on sentence-level syntactic structures and tailored to EAE task. Furthermore, we propose a novel regularization technique to explicitly constrain the contributions of unrelated context words in the final prediction for EAE. We perform extensive experiments on the benchmark document-level EAE dataset RAMS that leads to the state-of-the-art performance. Moreover, our experiments on the ACE 2005 dataset reveals the effectiveness of the proposed model in the sentence-level EAE by establishing new state-of-the-art results.

1 Introduction

Event Extraction (EE) is one of the important sub-tasks of Information Extraction (IE). The major goal of EE is to identify events and their engaged entities (i.e., event arguments). To this end, two sub-tasks should be solved: 1) Event Detection (ED): To recognize event triggers (i.e., the words or phrases that clearly specify the occurrence of events) and their event types, 2) Event Argument Extraction (EAE): To identify participants of events (i.e., entities engaged in events) and their argument roles. Compared to ED, the EAE sub-task is relatively less explored in IE. Moreover, prior works on EAE are mainly restricted to sentence-level setting where event triggers and arguments are assumed to appear in the same sentences. This is unfortunate as a considerable portion of event arguments might not be immediately mentioned in the same sentence as their event triggers. For instance, in the EE dataset of the DARPA AIDA program (phase 1) extraction[1], 38% of arguments has been shown to be outside sentences containing the corresponding triggers, i.e., in the document-level context (Ebner et al., 2020). Thus, further research to study the more realistic setting of document-level EAE is extremely needed.

To the best of our knowledge, there are two prior works on document-level EAE (Ebner et al., 2020; Chen et al., 2020). Both works employ representation vectors for argument spans obtained from a transformer to compute likelihood scores for argument roles. Unfortunately, those prior works only exploit the sequential order of words in documents to represent the arguments and totally ignore structures of input documents. This is a limitation as in other related document-level tasks, e.g., relation extraction (RE), document structures (i.e., graphs to capture interactions between different words/sentences) have been showed to enhance representation learning (Thayaparan et al., 2019; Gupta et al., 2019; Sahu et al., 2019; Christopoulou et al., 2019; Nan et al., 2020). In addition, syntactic structures (i.e., dependency trees) have not been exploited in prior document-level EAE models. To address those limitations, we aim to devise a deep learning model to effectively employ syntactic structures of input documents to boost the performance of EAE.

Given a document, how can we efficiently induce its syntactic structure? One simple solution, as demonstrated in (Gupta et al., 2019) for document-level RE, is to employ the syntactic structure (i.e., dependency tree) of each sentence and connect their roots to each other to create a connected graph

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

Findings of the Association for Computational Linguistics: ACL 2022, pages 1648 - 1658

for an input document (called document structure). However, this approach might introduce unrelated words for the role prediction of a candidate argument for a given event trigger into the graph, thus hindering effective representation learning for document-level context. To alleviate this issue, (Gupta et al., 2019) proposes to prune the document structure by only retaining words along the dependency path (DP) between the two words of interest (i.e., event trigger and argument candidate in our case). Unfortunately, in document-level EAE, related words for role predictions might not solely reside in the dependency path between the event trigger and argument candidate. In particular, some related words that belong to sentences other than the hosting sentences of the event trigger and argument might be excluded if the document structure is pruned along the dependency path. For instance, in the document "*The primary goal of the plan is to* ***provide*** *protection to refugees. According to reports, all 8* ***countries*** *that signed the plan will congregate once a quarter to monitor the progress.*", the trigger and the candidate argument, i.e., *provide* and *countries*, appear in different sentences and the DP between them is "*provide* $\rightarrow$ *is* $\rightarrow$ *congregate* $\rightarrow$ *countries*". However, in order to predict the role of the argument, i.e., *Giver*, one should consider the word *plan* in the first sentence and the words *plan* and *signed* in the second sentence which are off the DP.

These limitations call for better methods to prune dependency-based structures of documents to better preserve important words and exclude noisy ones. Unlike prior work that resorts to simple syntax-based rules, i.e., distance to the dependency path (Zhang et al., 2018), we argue that the pruning operation should be also aware of the semantics of the words. In other words, two criteria, i.e., syntactic and semantic relevance, should be taken into account. Specifically, a word is retained in the document structure for document-level EAE if it has a small distance to the event trigger/argument words in the dependency structure (i.e., syntax-based importance) and it is semantically related to one of the words in the dependency path (i.e., semantics-based importance). Note that the semantic similarity between words can be obtained from their representations induced by the model. A key challenge for this idea is the different nature of the syntactic and semantic distances that complicates the information combination to determine the importance of a word for the structure. In addition, the retention decision for a word should also be contextualized in the potential contributions of other words in the document structure for EAE. As such, motivated by the dependency path as the anchor for document structure pruning, we propose to cast the problem of joint consideration of syntactic and semantic distances of words into finding an optimal alignment between off-the-DP and on-the-DP words. The optimal alignment will be solved via Optimal Transport (OT) methods where syntactic and semantic distances of words to those on the dependency path are simultaneously modeled in a joint optimization problem. OT is an established mechanism to efficiently find an optimal transport plan (i.e., an alignment) between two groups of points (i.e., off-the-DP and on-the-DP words in our case) based on their pairwise transportation costs and the distribution mass accumulated on the points. We propose to employ semantic similarity of words to obtain their transportation costs while syntactic distances to the event trigger/argument are leveraged to compute the mass distributions of words for OT in our document-level EAE problem. Finally, to prune the document structure, an off-the-DP word is considered important for the document structure (thus being retained) if it is aligned to one of the on-the-DP words via the OT solution. The pruned document structure will be leveraged to learn representation vectors for input documents to perform argument role predictions using Graph Convolution Networks (GCN) (Kipf and Welling, 2017).

Although the OT-based pruning method could help exclude unrelated words for EAE in the document structure, their noisy information might be still encoded in the representations of the related words due to the contextualization in the input encoder (e.g., BERT). To improve the representation learning, we thus propose to explicitly constrain the impact of unrelated words for representation learning via a novel regularization technique based on the pruned document structure. In particular, we seek to add unrelated words back to the pruned structure (thus restoring the original tree) and ensure a minimized change of representation vectors due to this addition. As such, in addition to the pruned structure, we apply the GCN model over the original dependency structure to obtain another set of representations vectors for the words. Eventually, in the final loss function, we introduce the

difference between the representation vectors obtained from the pruned and original structures to achieve the contribution constraint for unrelated words. In our experiments, we evaluate our model on both sentence-level and document-level EAE benchmark datasets, demonstrating the effectiveness of the the proposed model by establishing new state-of-the-art results in both settings.

2 Model

Problem Definition: The goal of EAE task is to recognize the role of entity mentions toward a specific event trigger. We formulate this task as a multi-class classification problem. Formally, given a document $D = [w_1, w_2, \ldots, w_n]$, with the trigger word w_t and the candidate argument w_a, the goal is to predict one of the labels $L = [l_1, l_2, \ldots, l_m]$ as the role of the candidate argument w_a in the event evoked by the trigger w_t. The label set L contains a special label $None$ to indicate that the candidate argument w_a is not a participant in the event w_t.

Model Overview: The proposed model consists of four major components: 1) Input Encoder to represent the words in the document using high-dimensional vectors; 2) Dependency Pruning to employ Optimal Transport (OT) to prune unrelated words in the dependency tree; 3) Regularization to explicitly minimize the contribution of unrelated words for representation learning; and 4) Prediction to use the representations induced for the words of the document to make the final prediction.

2.1 Input Encoder

In the first step, we represent each word $w_i \in D$ using a high dimensional vector x_i. The vector x_i is constructed by concatenating the following vectors: A) Contextualized Word Embedding: We feed the input text $[CLS]w_1w_2 \ldots w_n[SEP]$ into the BERT$_{base}$ model (Devlin et al., 2019); we use the hidden state of w_i in the final layer as the contextualized word embedding. Note that for words consisting of multiple word-pieces, we take the average of its word-piece representations; and B) Distance Embeddings: We represent the relative distances of the word w_i toward the trigger and the argument words (i.e., $|i - t|$ and $|i - a|$) using high dimensional vectors obtained from a distance embedding table (initialized randomly). The distance embedding table is updated during training. Also, in our experiments, we find that fixing the BERT parameters is more helpful. As such, to tailor the vectors x_i to EAE task, we feed the vectors $X = [x_1, x_2, \ldots, x_n]$ to a Bi-directional Long Short-Term Memory network (BiLSTM). The hidden states obtained from the BiLSTM, $H = [h_1, h_2, \ldots, h_n]$, will be consumed by subsequent components.

2.2 Dependency Pruning

To employ the syntactic structure of the input document D, we leverage the dependency trees of the sentences in the document. Here, we use the undirected versions of the dependency trees generated by the Stanford CoreNLP parser. To connect the dependency trees of the sentences to form a single dependency graph for D, similar to (Gupta et al., 2019), we add an edge between the roots of the dependency trees for every pair of consecutive sentences in D. As such, the generated syntactic tree for D, called T, will contain all the words $w_i \in D$. As discussed in the introduction, the full tree T for D might contain both related and unrelated words for the argument role prediction of w_a with respect to the event trigger w_t. It is thus necessary to prune this tree to retain only the related words, thus preventing potential noises introduced by unrelated words for representation learning. Motivated by the effectiveness of dependency paths for sentence-level EAE in prior work (Li et al., 2013), we employ the dependency path (DP) between the event trigger w_t and the argument candidate w_a in T as the anchor to prune the unrelated words. In particular, besides the words along DP (that might miss some important context words for prediction), we seek to retrain only off-of-DP words in T that are syntactically and semantically close to to the words in DP (i.e., aligning off-of-DP and on-the-DP words). We propose to employ Optimal Transport (OT) to jointly consider syntax and semantics for this word alignment. In the following, we first formally describe OT. We will then provide details on how OT could be leveraged to implement our idea.

OT is an established method to find the optimal plan to convert (i.e., transport) one distribution to another one. Formally, given the probability distributions $p(x)$ and $q(y)$ over the domains $\mathcal{X}$ and $\mathcal{Y}$, and the cost/distance function $C(x, y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ for mapping $\mathcal{X}$ to $\mathcal{Y}$, OT finds the optimal joint alignment/distribution $\pi^*(x, y)$ (over $\mathcal{X} \times \mathcal{Y}$) with marginals $p(x)$ and $q(y)$, i.e., the cheapest transportation from $p(x)$ to

$q(y)$, by solving the following problem:

$$\pi^*(x,y) = \min_{\pi \in \Pi(x,y)} \int_{\mathcal{Y}} \int_{\mathcal{X}} \pi(x,y) C(x,y) dx dy \quad (1)$$
$$\textbf{s.t. } x \sim p(x) \text{ and } y \sim q(y),$$

where $\Pi(x,y)$ is the set of all joint distributions with marginals $p(x)$ and $q(y)$. Note that if the distributions $p(x)$ and $q(y)$ are discrete, the integrals in Equation 1 are replaced with a sum and the joint distribution $\pi^*(x,y)$ is represented by a matrix whose entry (x,y) ($x \in \mathcal{X}, y \in \mathcal{Y}$) represents the probability of transforming the data point x to y to convert the distribution $p(x)$ to $q(y)$. Note that to obtain a hard alignment between data points $\mathcal{X}$ and $\mathcal{Y}$, we can align each row of $\pi^*(x,y)$ with the column with the highest probability, i.e., $y^* = \text{argmax}_{y \in \mathcal{Y}} \pi^*(x,y)$ where y^* is the data point in $\mathcal{Y}$ aligned with the data point $x \in \mathcal{X}$.

The most important and useful characteristics of OT in our problem is that it can find a transportation (i.e., an alignment) between two groups of data points with lowest cost according to two criteria: 1) the distance between data points, and 2) the difference between their probability masses. In particular, these two criteria can be exploited to capture the semantic and syntactic similarity required in our model to find an alignment between off-the-DP and on-the-DP words. Specifically, we use the words on the DP as the data points in the domain $\mathcal{Y}$ and the words off the DP as the data points in the domain $\mathcal{X}$. To compute the distributions $p(x)$ and $q(y)$ (i.e., probability masses for data points) for $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, we use the syntax-based importance scores. Formally, for the word w_i, we compute its distance to the trigger and the candidate argument in the dependency tree (lengths of dependency paths), i.e., d_i^t and d_i^a, respectively. Afterward, the probability mass for a word $x = w_i \in \mathcal{X}$ is computed as the minimum of the two distances, i.e., $p(x) = \min(d_i^t, d_i^a)$. Note that the distribution $p(y)$ is computed similarly; $p(x)$ and $p(y)$ are also normalized with softmax over their corresponding sets to obtain distributions. In order to obtain the distance/transportation cost $C(x,y)$ between every pair of words $(x,y) \in \mathcal{X} \times \mathcal{Y}$, we propose to use their semantic information based on the Euclidean distance of their representation vectors h_x and h_y in H: $C(x,y) = \|h_x - h_y\|$.

Using this setup, solving Equation 1 returns the optimal alignment $\pi^*(x,y)$ that can be used to align each data point in $\mathcal{X}$ with one data point in $\mathcal{Y}$[2]. However, in our problem, we look for a subset of data points in $\mathcal{X}$ to be aligned with data points in $\mathcal{Y}$ for retention in the dependency structure for D. As such, we add an extra data point "*NULL*" to $\mathcal{Y}$ whose representation is computed by averaging the representations of all data points in $\mathcal{X}$ and probability mass is the average of probability masses of the data points in $\mathcal{X}$. Alignments with this data point in $\mathcal{Y}$ will serve as null alignment indicating that the aligned data point in $\mathcal{X}$, i.e., an off-the-DP word, should not be kept in the pruned tree. Other words in $\mathcal{X}$ with a non-null alignment, called $\mathcal{I}$ ($\mathcal{I} \subset \mathcal{X}$), will be preserved in the pruned tree for D. The removal of *NULL*-assigned off-of-DP words from T produces a new graph that presumably contains the most important words for argument role prediction for D. Here, to ensure the connectivity of the new graph, we also retain any words along the dependency paths between the trigger/argument words and a word in $\mathcal{I}$, leading to a new graph T' to represent D with important context words.

In the next step, we feed T' into a Graph Convolution Network (GCN) (Kipf and Welling, 2017; Nguyen and Grishman, 2018) to learn more abstract representation vectors for the words in T', leveraging BiLSTM-induced vectors in H as the inputs. We denote the hidden vectors produced in the last layer of the GCN model GCN by: $H' = h'_{i_1}, \ldots, h'_{i_m} = \text{GCN}(H, T')$ where m is the number of words in T' ($m < n$) and h'_{i_k} is the vector for the word w_{i_k} (i.e., the k-word in T').

2.3 Regularization

By using the pruned tree T' to compute the representation vectors in H', we expect to explicitly guide those vectors to: (i) encode related/important context words, and (ii) exclude potentially noisy information from unrelated words for the role prediction of w_a. However, due to the contextualization in the input encoder with BERT, the noisy information of unrelated words might still be included in the representations H for the selected words in the pruned tree T', thus being propagated by the GCN into the representations H'. As such, to further constrain the contribution of unrelated words for representation learning, we introduce a novel regularization technique that encourages

[2]Note that as solving the OT problem in Equation 1 is intractable, we employ the entropy-based approximation of OT and solve it with the Sinkhorn algorithm (Peyre and Cuturi, 2019).

the representations obtained from every word in D to be similar to the representations obtained only from the related words in T' (i.e., adding unrelated words does not change the representations significantly). As the output vectors from the GCN model will be used by the role prediction, we implement this regularization technique based on the representation vectors induced from GCN. Formally, we first feed the H and the full dependency tree T for D into the same GCN model, i.e., $H'' = \texttt{GCN}(H, T)$. Afterward, we compute the vector representation vectors $\bar{h}'$ and $\bar{h}''$ for the sets H' (based on T) and H'' (based on T) by performing a max-pooling, i.e., $\bar{h}' = MAX\_POOL(h'_{i_1}, \ldots, h'_{i_m})$ and $\bar{h}'' = MAX\_POOL(h''_1, h''_2, \ldots, h''_n)$. Finally, we enforce the similarity of $\bar{h}'$ and $\bar{h}''$ by adding their L_2 distance into the overall loss function: $\mathcal{L}_{reg} = \left\|\bar{h}' - \bar{h}''\right\|$.

2.4 Prediction

To perform the argument role prediction for w_a and w_t, we form the overall vector $V = [h'_t, h'_a, \bar{h}']$, where h'_t and h'_a are the representation vectors for w_a and w_t in H'. As such, V will be consumed by a two-layer feed-forward network to obtain the distribution $P(\cdot|D, w_t, w_a)$ over possible argument roles. To train the model, we use negative log-likelihood loss: $\mathcal{L}_{pred} = -\log P(l|D, w_t, w_a)$, where l is the gold label. The overall loss function for our model is thus: $\mathcal{L} = \mathcal{L}_{pred} + \beta\mathcal{L}_{reg}$, where β is a trade-off parameter.

3 Experiments

3.1 Datasets & Parameters

We evaluate the proposed model, i.e., Optimal Transport-based Event Argument Extraction (OTEAE), on the RAMS dataset which is recently introduced in (Ebner et al., 2020) for document-level EAE. RAMS contains 9,124 annotated event mentions across 139 event types for 65 argument roles, serving as the largest available dataset for this task. We use the official train/dev/test split and evaluation scripts for RAMS provided by (Ebner et al., 2020) for a fair comparison.

We use the development set of RAMS to fine tune the hyper parameters for the proposed model. Based on our experiments, the following hyper parameters are chosen: 50 dimensions for position embeddings; 1 layer for BiLSTM and 2 layers for GCN; 150 dimensions for the hidden states of the BiLSTM, GCN and feed-forward networks; 64 for the batch size; 0.2 for the learning rate with the Adam optimizer, and 0.1 for the trade-off parameter β.

3.2 Baselines

For the experiments on the RAMS dataset, we compare our model with two groups of baselines: (1) Prior works that report their performance on RAMS. Specifically, we compare our model with the **RAMS**$_{model}$ model in (Ebner et al., 2020), the **Head-based** model in (Zhang et al., 2020c) and the **Joint** model in (Chen et al., 2020). As such, the Joint model in (Chen et al., 2020) currently has the best reported performance on RAMS. Note that all of these baselines are sequence-based deep models that ignore the syntactic structure of the input document.; (2) To thoroughly compare OTEAE with previous works, we examine the structure-aware deep learning models proposed for another related task, i.e., document-level relation extraction, and adapt them for EAE. Concretely, we compare our model with: (i) the **iDepNN** model in (Gupta et al., 2019) that employs the syntactic structure of the document with pruning along the dependency path; (ii) the **GCNN** model in (Sahu et al., 2019) that leverages both syntactic and discourse-level (i.e., co-reference links) structures to encode the document; (iii) the **LSR** model in (Nan et al., 2020) that infers document structures by a deep reasoning module; and (iv) the **EoG** model in (Christopoulou et al., 2019) that encodes syntactic and discourse structures using high dimensional vectors to represent the edges of the structure graphs.

Following (Ebner et al., 2020), we report the performance of the models in two different settings for RAMS: (1) Standard Decoding: In this setting, the label is predicted by operating argmax on the probability distribution $P(\cdot|D, w_t, w_a)$; (2) Type Constrained: In this setting, the prediction of the models for a given candidate argument and event trigger is constrained to the set of permissible roles for the event type of the given event trigger. Specifically, before applying argmax on $P(\cdot|D, w_t, w_a)$, the probabilities of the non-permissible roles for the event type evoked by w_t are set to zero.

3.3 Performance Comparison

The performance of the models on the RAMS test set is presented in Table 1. As can be seen, the proposed model significantly outperforms both sequence-based and structure-aware baselines in both settings on the RAMS dataset (with $p < 0.01$),

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
Joint	-	-	-	79.6	80.2	79.9
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
OTEAE (ours)	75.2	76.1	**75.6**	83.1	78.9	**80.9**

Table 1: Performance on the RAMS test set. All models employ BERT for the input encoder.

Model	P	R	F1
BERT-based	57.9	59.1	58.5
GTM	62.1	64.3	63.2
Joint	56.0	79.2	65.6
OTEAE (ours)	64.2	68.1	**66.1**

Table 2: Sentence-level performance on the ACE 2005 test set.

yielding to the state-of-the-art performance for document-level EAE on RAMS. Compared to the sequence-based baselines (i.e., Head-based, Joint), we attribute the success of our model to the ability to capture long-distance dependencies between words in multiple sentences (via syntactic structures) that can encode documents with richer information. Moreover, compared to the document structure-aware baselines, our hypothesis for the superior performance of OTEAE involves the OT-based component that is able to recognize the optimal trade-off between semantics-based and syntax-based importance of the words to better filter unrelated words to learn document structures for EAE. In particular, most baseline models employ human-designed rules to compute document structures that cannot flexibly prune unrelated words e.g., iDepNN (Gupta et al., 2019) to prune syntactic structures along dependency paths, and EoG (Christopoulou et al., 2019) and GCNN (Sahu et al., 2019) to employ heuristic discourse information (coreference links), thus leading to inferior performance.

3.4 Performance on ACE 2005

To provide more insight into the performance of the proposed model OTEAE, following (Chen et al., 2020), we further examine the models' performance on the well-known ACE 2005 dataset for the sentence-level EAE task. This dataset contains 599 documents, 33 event subtypes and 35 argument roles. We employ the same data split and pre-processing scripts as prior works (Lin et al., 2020; Chen et al., 2020). To be directly comparable with (Chen et al., 2020) (the current state-of-the-art model for document-level EAE), we assume golden event trigger and argument spans in this experiment. Table 2 reports the performance of the models on the ACE 2005 test set. Here in addition to OTEAE and Joint (Chen et al., 2020), we show the performance of two other models: (i) the BERT-based model that directly uses the BiLSTM vectors in H to form the overall representation vector $V = [h_t, h_a, MAX\_POOL(h_1, \ldots, h_n)]$ for predictions (i.e., the OT-based pruning and regularization are not applied here); and (ii) the GTM model in (Veyseh et al., 2020) that currently has the current state-of-the-art model for sentence-level EAE on ACE 2005.

As can be seen from the table, OTEAE still performs well even with shorter context of single sentences, leading to state-of-the-art performance for sentence-level EAE. Notably, the substantially better performance of OTEAE over BERT-based shows that the proposed dependency pruning and regularization components are also beneficial for representation learning in sentence-level EAE.

3.5 Ablation Study

OTEAE has two major components: (1) The structure generator component to infer pruned dependency structures for documents, (2) The regularization component to explicitly exclude the unrelated information. This section conducts an ablation study to analyze the effectiveness of these components for OTEAE. In particular, we evaluate the performance of the following ablated models: (1) **Reg**$^-$: This model excludes the regularization loss, i.e., $\mathcal{L}_{reg}$, from the overall loss function $\mathcal{L}$; (2) **OT**$^-$: This baseline eliminates the OT-based component for tree pruning, instead, it prunes dependency structures along dependency paths; (3) **Prune**$^-$: This model employs the full dependency tree as the structure to be consumed by the GCN model. As such, the regularization component which requires a pruned tree is also excluded from the final loss function; (4) **GCN**$^-$: This model excludes the GCN model from OTEAE. Here, we still retain the OT-based pruning and regularization components; however, instead of using GCN-based representations, the vectors for final predictions and regularization need to be computed over the BiLSTM-induced vectors in H. In particular, the final prediction vec-

Model	Precision	Recall	F1
OTEAE (full)	74.9	75.5	**75.2**
Reg$^-$	74.3	73.9	74.1
OT$^-$	72.8	73.2	73.0
Prune$^-$	72.3	72.0	72.2
GCN$^-$	72.2	71.5	71.9

Table 3: Ablation study on the RAMS development set.

Model	Precision	Recall	F1
OTEAE (full)	74.9	75.5	**75.2**
Syntax$^-$	75.1	73.5	74.3
Semantics$^-$	74.4	73.5	73.9
DP$^-$	74.1	73.1	73.6

Table 4: Optimal Transport analysis on the RAMS development set.

tor V is constructed as $V = [h_t, h_a, \hat{h}]$ where $\hat{h} = MAX\_POOL(h_{i_1}, \ldots, h_{i_m})$ (i.e., the max-pooling is done over the words in the pruned tree T' from OT); the regularization term in the overall loss function is replaced by: $\mathcal{L}_{reg} = \left\|\hat{h} - \tilde{h}\right\|$ where $\tilde{h} = MAX\_POOL(h_1, \ldots, h_n)$ (max-pooling is performed over all the words in D). Results of this analysis are shown in Table 3. This table demonstrates the necessity of all components for the proposed model to achieve its highest performance. In particular, the superior performance of OTEAE over OT$^-$ and Prune$^-$ suggests that using only dependency paths or full dependency structures is suboptimal to produce document structures for document-level EAE, necessitating OT to better select important context words for documents in our problem.

In order to provide more insight into the importance of OT for tree pruning, we perform another ablation study solely on the OT component. Specifically, we answer two questions: (1) Are both syntax-based and semantic-based criteria necessary to prune the dependency tree? (2) Should the dependency paths be taken into account during pruning? For the first question, we study two ablated models: (1) **Syntax$^-$**: In this model, we use uniform distribution for $p(x)$ and $q(y)$ in OT, thus excluding the syntactic distances of the words to the trigger/argument from OT computation; (2) **Semantics$^-$**: In this baseline, we use a constant cost function, i.e., $C(x, y) = 1$, for OT so the representation-based similarities between the words are not used by OT. To answer the second question, we evaluate the model **DP$^-$** where the domain (Y) only consists of the trigger and the argument words; and the domain $\mathcal{X}$ involves all other words in D (including the ones on the dependency paths). Note that in this model, we still add the extra node "*NULL*" into $\mathcal{Y}$ to represent null alignments. Table 4 presents the performance of the models. There are several observations from this table. First, removing either syntax-based (i.e., Syntax$^-$) or semantic-based (i.e., Semantics$^-$) criterion will hurt the performance, indicating the necessity of both criteria. Second, compared to the syntax-based information, the semantic-based criterion contributes more to OTEAE as removing it will lead to larger performance reduction. This is important as the semantic-based criterion has not been used in prior methods for document structure inference with tree pruning. Finally, using only the trigger/argument words as the anchor points for positive alignment (i.e., DP$^-$) is not optimal, showing that dependency paths are critical for OT to find related words in documents for EAE.

4 Analysis

Inter- vs. Intra-sentence Performance: To shed more light on the effectiveness of the proposed model, we study the performance of the proposed model in two different settings: (1) Intra-sentence where both trigger and argument words appear in the same sentence, i.e. the number of sentences between the trigger and the argument is zero; and (2) Inter-sentence where the trigger and argument appear in different sentences, i.e., the number of sentences in between is non-zero. We compare our model with the state-of-the-art models for document-level EAE, i.e., the RAMS$_{model}$ (Ebner et al., 2020) and Joint (Chen et al., 2020) models in this analysis. All models assumes Type-Constrained decoding in this section. The results on the RAMS development set are shown in Table 6. This table shows that the proposed model significantly outperforms prior works (except for the distance of 2 where the performance is comparable). Interestingly, the obtained improvement is consistent for both inter-sentence and intra-sentence settings, suggesting the benefits of using efficient document structure for EAE in general.

Case Study: Finally, we perform a case study to explore the benefits of OTEAE compared to prior document-level methods. In particular, we analyze examples in which our model successfully predicts the argument role, while all other docu-

ID	Text	Role
1	The *massive explosions* destroyed vehicles on a highway just outside the *base at the Syrian port-city* of Tartus, northwestern Syria. It is understood the *first blast* was a car bomb planetout outside the *base*. The *second explosion* was a suicide bomber who **detonated** his belt as people rushed to help those injured, AFP reported.	Place
2	There are worrying reports of the *tundra burning* in the Arctic Yamal Peninsula, as well as other damaging *fires*, for example a 3,000 hectare **blaze** at the Lena Pillars Nature Park. Ecologists say the *fires* pose a direct *threat* to the *role* of Siberian pristine Boreal in absorbing climate-warming emissions.	Instrument

Table 5: Case study on the RAMS development set. Event triggers are shown in **red bold-face**; arguments is shown in blue underlined font; and the OT-selected words in OTEAE are presented in green italic font.

Dist.	# Gold args.	RAMS	Joint	Ours
-2	79	75.7	77.2	**78.1**
-1	164	73.7	74.4	**75.2**
0	1,811	75.0	79.6	**80.3**
1	87	76.5	77.0	**77.5**
2	47	**79.1**	78.7	79.0

Table 6: F1 scores on the RAMS development set for examples with different numbers of sentences between the trigger and the argument. Negative numbers imply that the argument appears before the trigger.

ment structure-aware baselines fail (i.e., iDepNN, EoG, GCNN, and LSR). Some examples of this type are shown in Table 5. In particular, for the first example (ID 1), the trigger and the argument are in two different sentences with an extra sentence in between. Due to the long distance between the trigger and the argument, using the document structure is crucial to infer the role of the argument. Specifically, a successful prediction should encode the mentions of *massive explosions* in the first sentence and *second explosion* in the second sentence, and their semantic similarity. Unfortunately, none of these phrases are on the DP between the trigger and the argument in the document's dependency graph, causing the failure of the baseline models. In contrast, the OT-based selection method in OTEAE is able to select both phrases to include in the pruned tree T' for representation learning, thus being able to make a correct prediction.

Finally, in the second example document (ID 2) of Table 5 with two sentences, to correctly predict the argument role for "*Sibreian pristine Boreal*", it is important to consider the word *fire* in the second sentence. Unfortunately this word does not belong to the dependency path between the trigger and argument, hindering the operation of prior models for this example. OTEAE, in contrast, can return a correct prediction in this case as the its OT component helps include the important word "*fire*" into consideration for the document structure.

5 Related Works

Event Argument Extraction is one of the sub-tasks of Event Extraction which is mainly approached with sentence-level models in the prior works. Early models for this task employed feature-based methods (Patwardhan and Riloff, 2009; Riedel and McCallum, 2011; Hong et al., 2011; McClosky et al., 2011; Li et al., 2013; Miwa et al., 2014; Yang and Mitchell, 2016). Later, deep learning emerged as the state-of-the-art approach for sentence-level EE (Chen et al., 2015; Zhang et al., 2019; Yang et al., 2019; Nguyen and Nguyen, 2019; Zhang et al., 2020b; Lai et al., 2020; Nguyen et al., 2021; Veyseh et al., 2021) or specifically, EAE (Wang et al., 2019). As the sentence-level models are not able to detect all arguments of an event mentioned in a document, recently the document-level setting has gained more attention; a new dataset (i.e., RAMS) has been introduced by (Ebner et al., 2020). The most similar models to our document-level EAE model involve RAMS$_{model}$ (Ebner et al., 2020), Head-based (Zhang et al., 2020c) and the Joint model (Chen et al., 2020). However, none of these models explore document structures for representation that have been shown to be critical for EAE in our work.

Document structures has been also exploited for other Information Extraction (Sahu et al., 2019; Gupta et al., 2019; Nan et al., 2020; Tran et al., 2020) and NLP tasks (Pan et al., 2020; Balachandran et al., 2020; Zhang et al., 2020a; Lu et al., 2021). In particular, for the related task of document-level relation extraction, existing works have attempted to construct document structures based on the syntax or discourse information. However, these models fail to involve semantics of the document in the constructed structure, resulting in the inferior performance. In this work, we address this limitation by exploiting optimal transport to jointly consider syntactic and semantic information for document structures. To our knowledge, OT has not been used for document structures in prior work.

6 Conclusion

In this work, we propose a new document structure-aware model for document-level EAE. Our model employs dependency trees of sentences and presents a novel technique based on optimal

transport to prune dependency trees for documents in the EAE task. In addition, we introduce a novel regularization to explicitly constrain the contribution of irrelevant words for representation learning. Our extensive experiments demonstrate the effectiveness of the proposed model. In the future, we plan to apply our model for other IE tasks.

Acknowledgments

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112 and the NSF grant CNS-1747798 to the IU-CRC Center for Big Learning. 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 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 ARO, ODNI, IARPA, the Department of Defense, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

References

Vidhisha Balachandran, Artidoro Pagnoni, Jay Yoon Lee, Dheeraj Rajagopal, Jaime Carbonell, and Yulia Tsvetkov. 2020. Structsum: Incorporating latent and explicit sentence dependencies for single document summarization. In *arXiv preprint arXiv:2003.00576*.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yunmo Chen, Tongfei Chen, and Benjamin Van Durme. 2020. Joint modeling of arguments for event understanding. In *Proceedings of the First Workshop on Computational Approaches to Discourse*.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Pankaj Gupta, Subburam Rajaram, Hinrich Schütze, and Thomas Runkler. 2019. Neural relation extraction within and across sentence boundaries. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

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

Qiuhao Lu, Thien Huu Nguyen, and Dejing Dou. 2021. Predicting patient readmission risk from medical text via knowledge graph enhanced multiview graph convolution. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.

David McClosky, Mihai Surdeanu, and Christopher Manning. 2011. Event extraction as dependency parsing. In *BioNLP Shared Task Workshop*.

Makoto Miwa, Paul Thompson, Ioannis Korkontzelos, and Sophia Ananiadou. 2014. Comparable study of event extraction in newswire and biomedical domains. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Minh Van Nguyen, Viet Dac Lai, and Thien Huu Nguyen. 2021. Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Liangming Pan, Yuxi Xie, Yansong Feng, Tat-Seng Chua, and Min-Yen Kan. 2020. Semantic graphs for generating deep questions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Gabriel Peyre and Marco Cuturi. 2019. Computational optimal transport: With applications to data science. In *Foundations and Trends in Machine Learning*.

Sebastian Riedel and Andrew McCallum. 2011. Robust biomedical event extraction with dual decomposition and minimal domain adaptation. In *BioNLP Shared Task 2011 Workshop*.

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 Annual Meeting of the Association for Computational Linguistics (ACL)*.

Mokanarangan Thayaparan, Marco Valentino, Viktor Schlegel, and André Freitas. 2019. 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*.

Hieu Minh Tran, Minh Trung Nguyen, and Thien Huu Nguyen. 2020. The dots have their values: Exploiting the node-edge connections in graph-based neural models for document-level relation extraction. In *Proceedings of the Findings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Amir Pouran Ben Veyseh, Viet Dac Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2021. Unleash gpt-2 power for event detection. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

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 Findings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xiaozhi Wang, Ziqi Wang, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, Maosong Sun, Jie Zhou, and Xiang Ren. 2019. Hmeae: Hierarchical modular event argument extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Junchi Zhang, Yanxia Qin, Yue Zhang, Mengchi Liu, and Donghong Ji. 2019. Extracting entities and events as a single task using a transition-based neural model. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. 2020a. Every document owns its structure: Inductive text classification via graph neural networks. In *arXiv preprint arXiv:2004.13826*.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yunyan Zhang, Guangluan Xu, Yang Wang, Daoyu Lin, Feng Li, Chenglong Wu, Jingyuan Zhang, and Tinglei Huang. 2020b. A question answering-based framework for one-step event argument extraction. In *IEEE Access, vol 8, 65420-65431*.

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020c. A two-step approach for implicit event argument detection. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.