Exploiting Document Structures and Cluster Consistencies for Event Coreference Resolution

Hieu Minh Tran[1], **Duy Phung**[1] **and Thien Huu Nguyen**[2]
[1] VinAI Research, Vietnam
[2] Department of Computer and Information Science, University of Oregon,
Eugene, OR 97403, USA
{v.hieutm4,v.duypv1}@vinai.io, thien@cs.uoregon.edu

Abstract

We study the problem of event coreference resolution (ECR) that seeks to group coreferent event mentions into the same clusters. Deep learning methods have recently been applied for this task to deliver state-of-the-art performance. However, existing deep learning models for ECR are limited in that they cannot exploit important interactions between relevant objects for ECR, e.g., context words and entity mentions, to support the encoding of document-level context. In addition, consistency constraints between golden and predicted clusters of event mentions have not been considered to improve representation learning in prior deep learning models for ECR. This work addresses such limitations by introducing a novel deep learning model for ECR. At the core of our model are document structures to explicitly capture relevant objects for ECR. Our document structures introduce diverse knowledge sources (discourse, syntax, semantics) to compute edges/interactions between structure nodes for document-level representation learning. We also present novel regularization techniques based on consistencies of golden and predicted clusters for event mentions in documents. Extensive experiments show that our model achieve state-of-the-art performance on two benchmark datasets.

1 Introduction

Event coreference resolution (ECR) is the task of clustering event mentions (i.e., trigger words that evoke an event) in a document such that each cluster represents a unique real world event. For example, the three event mentions in Figure 1, i.e., "*refuse to sign*, "*raised objections*", and "*doesn't sign*", should be grouped into the same cluster to indicate their coreference to the same event.

A common component in prior ECR models involves a binary classifier that receives a pair of event mentions and predict their coreference (Chen et al., 2009; Lu et al., 2016; Lu and Ng, 2017). To this end, an important step in ECR models is to transform event mention pairs into representation vectors to encode discriminative features for coreference prediction. Early work on ECR has achieved feature representation via feature engineering where multiple features are hand-designed for input event mention pairs (Lu and Ng, 2017). A major problem with feature engineering is the sparsity of the features that limits the generalization to unseen data. Representation learning in deep learning models has recently been introduced to address this issue, leading to more robust methods with better performance for ECR (Nguyen et al., 2016; Choubey and Huang, 2018; Huang et al., 2019; Barhom et al., 2019). As such, there are at least two limitations in existing deep learning models for ECR that will be addressed in this work to improve the performance.

First, as event mentions pairs for coreference prediction might belong to long-distance sentences in documents, capturing document-level context between the event mentions (i.e., beyond the two sentences that host the event mentions) might present useful information for ECR. As their first limitation, prior deep learning models for ECR has only attempted to encode document-level context via hand-designed features (Kenyon-Dean et al., 2018; Barhom et al., 2019) that still suffer from the feature sparsity issue. In addition, such prior work is unable to exploit ECR-related objects in documents (e.g., entity mentions, context words) and their connections/interactions (possibly beyond sentence boundary) to aid representation learning. An example for the importance of context words, entity mentions, and their interactions for ECR can be seen in Figure 1. Here, to decisively determine the coreference of "*raised objections*" and "*doesn't sign*", ECR systems should recognize "*Trump*" and "*the*

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 4840–4850
August 1–6, 2021. ©2021 Association for Computational Linguistics

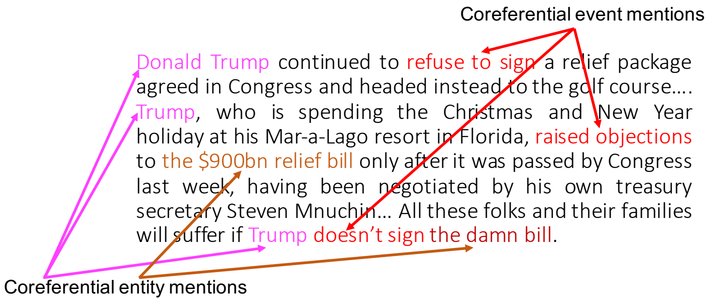
Coreferential event mentions
Donald Trump continued to refuse to sign a relief package agreed in Congress and headed instead to the golf course…. Trump, who is spending the Christmas and New Year holiday at his Mar-a-Lago resort in Florida, raised objections to the $900bn relief bill only after it was passed by Congress last week, having been negotiated by his own treasury secretary Steven Mnuchin… All these folks and their families will suffer if Trump doesn't sign the damn bill.
Coreferential entity mentions

Figure 1: An example for event coreference resolution.

$900bn relief bill" as the arguments of "*raised objections*", and "*Trump*" and "*the damn bill*" as the arguments of "*doesn't sign*". The systems should also be able to realize the coreference relation between the two entity mentions "*Trump*", and between "*the $900bn relief bill*" and "*the damn bill*" to conclude the same identity for the event mentions (i.e., as they involve the same arguments). As such, it is helpful to identify relevant entity mentions, context words and leverage their relations/interactions to improve representation vectors for event mentions in ECR. Motivated by this issue, we propose to form graphs for documents (called document structures) to explicitly capture relevant objects and interactions for ECR that will be consumed to learn representation vectors for event mentions. In particular, context words, entity mentions, and event mentions will serve as the nodes in our document structures due to their intuitive relevance to ECR. Different types of knowledge sources will then be exploited to connect the nodes for the document structures, featuring discourse information (e.g., to connect coreferring entity mentions), syntactic information (e.g., to directly link event mentions and their arguments), and semantic similarity (e.g., to connect words/event mentions with similar meanings). Such rich document structures allows us to model the interactions of relevant objects for ECR beyond sentence level for document-level context. Using graph convolutional neural networks (GCN) (Kipf and Welling, 2017; Nguyen and Grishman, 2018) for representation learning, we expect enriched representation vectors from the document structures can further improve the performance of ECR systems. To our knowledge, this is the first time that rich document structures are employed for ECR.

Second, prior deep learning models for ECR fails to leverage consistencies between golden clusters (provided by human) and predicted clusters (generated by models) to promote representation learning. In particular, it is intuitive that ECR models can achieve better performance if their predicted event clusters are more similar to the golden event clusters in the data. To this end, we propose to obtain different inconsistency measures between golden and predicted clusters that will be incorporated into the overall loss function for minimization. As such, we expect that the consistency/similarity regularization between two types of clusters can provide useful training signals to improve representation vectors for event mentions in ECR. To our knowledge, this is also the first work to exploit cluster consistency-based regularization for representation learning in ECR. Finally, we conduct extensive experiments for ECR on the KBP benchmark datasets. The experiments demonstrate the benefits of the proposed methods and lead to state-of-the-art performance for ECR.

2 Related Work

Event coreference resolution is broadly related to works on entity coreference resolution that aim to resolve nouns phrases/mentions for entities (Raghunathan et al., 2010; Ng, 2010; Durrett and Klein, 2013; Lee et al., 2017a; Joshi et al., 2019b,a). However, resolving event mentions has been considered as a more challenging task than entity coreference resolution due to the more complex structures of event mentions (Yang et al., 2015).

Our work focuses on the within-document setting for ECR where input event mentions are expected to appear in the same input documents; however, we also note prior works on cross-document ECR (Lee et al., 2012a; Adrian Bejan and Harabagiu, 2014; Choubey and Huang, 2017; Kenyon-Dean et al., 2018; Barhom et al., 2019; Cattan et al., 2020). As such, for within-document

ECR, previous methods have applied feature-based models for pairwise classifiers (Ahn, 2006; Chen et al., 2009; Cybulska and Vossen, 2015; Peng et al., 2016), spectral graph clustering (Chen and Ji, 2009), information propagation (Liu et al., 2014), markov logic networks (Lu et al., 2016), joint modeling of ECR with event detection (Araki and Mitamura, 2015; Lu et al., 2016; Chen and Ng, 2016; Lu and Ng, 2017), and recent deep learning models (Nguyen et al., 2016; Choubey and Huang, 2018; Huang et al., 2019; Lu et al., 2020; Choubey et al., 2020). Compared to previous deep learning works for ECR, our model presents a novel representation learning framework based on document structures to explicitly encode important interactions between relevant objects, and representation regularization to exploit the cluster consistency between golden and predicted clusters for event mentions.

3 Model

Formally, in ECR, given an input document $D = w_1, w_2, \dots, w_N$ (of N words/tokens) with a set of event mentions $E = \{e_1, e_2, \dots, e_{|E|}\}$, the goal is to group the event mentions in E into clusters to capture the coreference relation between mentions. Our ECR model consists of four major components: (i) Document Encoder to words into representation vectors, (ii) Document Structure to create graphs for documents and learn rich representation vectors for event mentions, (iii) End-to-end Resolution to simultaneously resolve the coreference for the entity mentions in D, and (iv) Cluster Consistency Regularization to regularize representation vectors based on consistency constraints between golden and predict event mention clusters. Figure 2 presents an overview of our model for ECR.

3.1 Document Encoder

In the first step, we transform each word $w_i \in D$ into a representation vector x_i by feeding D into the pre-trained language model BERT (Devlin et al., 2019). In particular, as BERT might split w_i into several word-pieces, we average the hidden vectors of the word-pieces of w_i in the last layer of BERT to obtain the representation vector x_i for w_i. To handle long documents with BERT, we divide D into segments of 512 consecutive word-pieces that will be encoded separately. The resulting sequence $X = x_1, x_2, \dots, x_n$ for D is then sent to the next steps for further computation.

3.2 Document Structure

This component aims to learn representation vectors for the event mentions in E using an interaction graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ for D that facilitates the enrichment of representation vectors for event mentions with relevant objects and interactions at document level. As such, the nodes and edges in $\mathcal{G}$ for our ECR problem are constructed as follows:

Nodes: The node set $\mathcal{N}$ for our interaction graph $\mathcal{G}$ should capture relevant objects for the coreference between event mentions in D. Toward this goal, we consider all the context words (i.e., w_i), event mentions, and entity mentions in D as relevant objects for our ECR problem. For convenience, let $M = \{m_1, m_2, \dots, m_{|M|}\}$ be the set of entity mentions in D. The node set $\mathcal{N}$ for $\mathcal{G}$ is thus created by the union of D, E, and M: $\mathcal{N} = D \cup E \cup M = \{n_1, n_2, \dots, n_{|\mathcal{N}|}\}$. To achieve a fair comparison, we use the predicted event mentions that are provided by (Choubey and Huang, 2018) in the datasets for E. The Stanford CoreNLP toolkit is employed to obtain the entity mentions M.

Edges: The edges between the nodes in $\mathcal{N}$ for $\mathcal{G}$ will be represented by an adjacency matrix $A = \{a_{ij}\}_{i,j=|\mathcal{N}|}$ ($a_{ij} \in \mathbb{R}$) in this work. As A will be consumed by Graph Convolutional Networks (GCN) to learn representation vectors for ECR, the value/score a_{ij} between two nodes n_i and n_j in $\mathcal{N}$ is expected to estimate the importance (or the level of interaction) of n_j for the representation computation of n_i. This structure allows n_i and n_j of $\mathcal{N}$ to directly interact and influence the representation computation of each other even if they are sequentially far away from each other in D. As presented in the introduction, we explore three types of information to design the edges $\mathcal{E}$ (or compute the interaction scores a_{ij}) for $\mathcal{G}$ in our model, including discourse-based, syntax-based and semantic-based information.

Discourse-based Edges: Due to multiple sentences and event/entity mentions involved in the input document D, we need to understand where such objects span and how they relate to each other to effectively encode document context for ECR. To this end, we propose to exploit three types of discourse information to obtain the interaction graph $\mathcal{G}$, i.e., sentence boundary, coreference structure, and mention span for event/entity mentions in D.

Sentence Boundary: Our motivation for this information is that event/entity mentions appearing

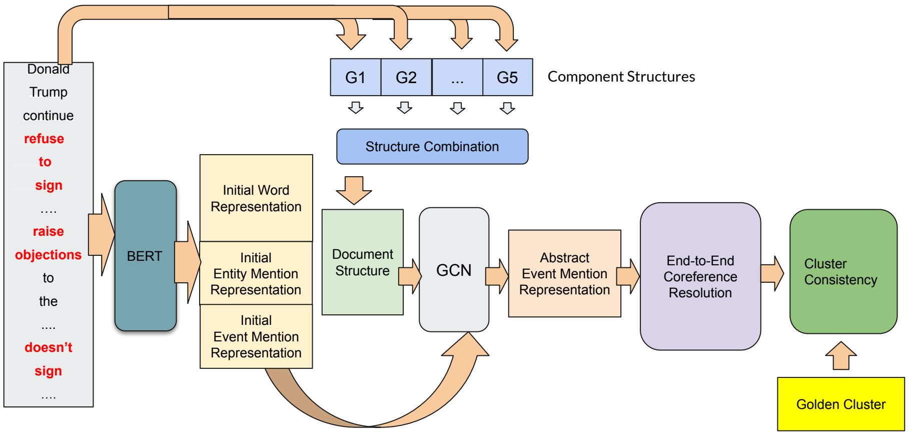

Figure 2: An overview of the proposed ECR model.

in the same sentences tend to be more contextually related to each other than those in different sentences. As such, event/entity mentions in the same sentences might involve more helpful information for the representation computation of each other in our problem. To capture this intuition, we compute the sentence boundary-based interaction score a_{ij}^{sent} for the nodes n_i and n_j in $\mathcal{N}$ where $a_{ij}^{sent} = 1$ if n_i and n_j are the event/entity mentions of the same sentences in D (i.e., $n_i, n_j \in E \cup M$); and 0 otherwise. We will use a_{ij}^{sent} as an input to compute the overall interaction score a_{ij} for $\mathcal{G}$ later.

Entity Coreference Structure: Instead of considering within-sentence information as in a_{ij}^{sent}, coreference structure focuses on the connection of entity mentions across sentences to enrich their representations with the contextual information of the coreferring ones. As such, to enable the interaction of representations for coreferring enity mentions, we compute the conference-based score a_{ij}^{coref} for each pair of nodes n_i and n_j to contribute to the overall score a_{ij} for representation learning. Here, a_{ij}^{coref} is set to 1 if n_i and n_j are coreferring entity mentions in D, and 0 otherwise. Note that we also use the Stanford CoreNLP toolkit to determine the coreference of entity mentions in this work.

Mention Span: The sentence boundary and coreference structure scores model interactions of event and entity mentions in D based on discourse structure. To connect event and entity mentions to context words w_i for representation learning, we employ the mention span-based interaction score a_{ij}^{span} as another input for a_{ij}. Here, a_{ij}^{span} is only set to 1 (i.e., 0 otherwise) if n_i is a word ($n_i \in D$) in the span of the entity/event mention n_j ($n_j \in E \cup M$) or vice verse. a_{ij}^{span} is important as it helps ground representation vectors of event/entity mentions to the contextual information in D.

Syntax-based Edges: We expect the dependency trees of the sentences in D to provide beneficial information to connect the nodes in $\mathcal{N}$ for effective representation learning in ECR. For example, dependency trees have been used to retrieve important context words between an event mentions and their arguments in prior work (Li et al., 2013; Veyseh et al., 2020a,b). To this end, we propose to employ the dependency relations/connections between the words in D to obtain a syntax-based interaction score a_{ij}^{dep} for each pair of nodes n_i and n_j in $\mathcal{N}$, serving as an additional input for a_{ij}. In particular, by inheriting the graph structures of the dependency trees of the sentences in D, we set a_{ij}^{dep} to 1 if n_i and n_j are two words in the same sentence (i.e., $n_i, n_j \in D$) and there is an edge between them in the corresponding dependency tree[1], and 0 otherwise.

Semantic-based Edges: This information leverages the semantic similarity of the nodes in $\mathcal{N}$ to enrich the overall interaction scores a_{ij} for $\mathcal{G}$. Our motivation is that a node n_i will contribute more to the representation computation of another node n_j for ECR if n_i is more semantically related to n_j. In particular, as the representation vectors for the nodes in $\mathcal{N}$ have captured the contextual semantics of the words in D, we propose to explore

[1] We use Stanford CoreNLP to parse sentences.

a novel source of semantic information that relies on external knowledge for the words to compute interaction scores between the nodes $\mathcal{N}$ in our document structures for ECR. We expect the external knowledge for the words to provide complementary information to the contextual information in D, thus further enriching the overall interaction scores a_{ij} for the nodes in $\mathcal{N}$. To this end, we propose to utilize WordNet (Miller, 1995), a rich network of word meanings, to obtain external knowledge for the words in D. The word meanings (i.e., synsets) in WordNet are connected to each other via different semantic relations (e.g., synonyms, hyponyms). In particular, our first step to generate knowledge-based similarity scores involves mapping each word node $n_i \in D \cap \mathcal{N}$ to a synset node M_i in WordNet using a Word Sense Disambiguation (WSD) tool. In particular, we employ WordNet 3.0 and the state-of-the-art BERT-based WSD model in (Blevins and Zettlemoyer, 2020) to perform the word-synset mapping in this work. Afterward, we compute a knowledge-based similarity score a_{ij}^{struct} for each pair of word nodes n_i and n_j in $D \cap \mathcal{N}$ using the structure-based similarity of their linked synsets M_i and M_j in WordNet (i.e., $a_{ij}^{struct} = 0$ if either n_i or n_j is not a word node in $D \cap \mathcal{N}$). Accordingly, the Lin similarity measure (Lin et al., 1998) for synset nodes in WordNet is utilized for this purpose: $a_{ij}^{struct} = \frac{2*\text{IC}(\text{LCS}(M_i,M_j))}{\text{IC}(M_i)+\text{IC}(M_j)}$. Here, IC and LCS represent the information content of synset nodes and the least common subsumer of two synsets in the WordNet hierarchy (the most specific ancestor node) respectively[2].

Structure Combination: Up to now, five scores have been generated to capture the level of interactions in representation learning for each pair of nodes n_i and n_j in $\mathcal{N}$ according to different information sources (i.e., $a_{ij}^{sent}, a_{ij}^{coref}, a_{ij}^{span}, a_{ij}^{dep}$ and a_{ij}^{struct}). For convenience, we group the five scores for each node pair n_i and n_j into a vector $d_{ij} = [a_{ij}^{sent}, a_{ij}^{coref}, a_{ij}^{span}, a_{ij}^{dep}, a_{ij}^{struct}]$ of size 5. To combine the scores in d_{ij} into an overall rich interaction score a_{ij} for n_i and n_j in $\mathcal{G}$, we use the following normalization:

$$a_{ij} = \exp(d_{ij}q^T)/\sum_{u=1..|\mathcal{N}|} \exp(d_{iu}q^T) \qquad (1)$$

where q is a learnable vector of size 5.

Representation Learning: Given the combined interaction graph $\mathcal{G}$ with the adjacency matrix $A = \{a_{ij}\}_{i,j=|\mathcal{N}|}$, we use GCNs to induce representation vectors for the nodes in $\mathcal{N}$ for ECR. In particular, our GCN model takes the initial representation vectors v_i of the nodes $n_i \in \mathcal{N}$ as the input. Here, the initial representation vector v_i for a word node $n_i \in D$ is directly obtained from the BERT-based representation vector $x_c \in X$ (i.e., $v_i = x_c$) of the corresponding word w_c for n_i. In contrast, for event and entity mentions, their initial representation vectors are obtained by max-pooling the contextualized embedding vectors in X that correspond to the words in the event/entity mentions' spans. For convenience, we organize v_i into rows of the input matrix $H_0 = [v_1, \dots, v_{|\mathcal{N}|}]$. The GCN model then involves G layers that generate the matrix H_l at the l-th layer for the nodes in $\mathcal{N}$ ($1 \le l \le G$) via: $H_l = ReLU(AH_{l-1}W_l)$ (W_l is the weight matrix for the l-th layer). The output of the GCN model after G layers is H_G whose rows are denoted by $H_G = [h_1, \dots, h_{|\mathcal{N}|}]$, serving as more abstract representation vectors for the nodes n_i in the coreference prediction for event mentions. Also, for convenience, let $\{r_{e_1}, \dots, r_{e_{|E|}}\} \subset H_G$ be the set of GCN-induced representation vectors for the event mention nodes in $e_1, \dots, e_{|E|}$ in E.

3.3 End-to-end Coreference Resolution

To facilitate the incorporation of the consistency regularization between golden and predicted clusters into the training process, we perform and end-to-end procedure that seeks to simultaneously resolve the coreference for the event mentions in E in a single process. Motivated by the entity coreference resolution in (Lee et al., 2017b), we implement the end-to-end resolution via a set of antecedent assignments for the event mentions in E. In particular, we assume that the event mentions in E are enumerated in their appearance order in D. As such, our model aims to link each event mention $e_i \in E$ to one of its prior event mention in the set $\mathcal{Y}_i = \{\epsilon, e_1, \dots, e_{i-1}\}$ (ϵ is a dumpy antecedent). Here, a link of e_i to a non-dumpy antecedent e_j in Y_i represents a coreference relation between e_i and e_j. In contrast, a dumpy assignment for e_i indicates that e_i is not coreferent with any prior event mention. By forming a coreference graph with e_i as the nodes, the non-dumpy antecedent assignments for every event mention in E can be utitlized to

[2] We use the nltk tool to obtain the Lin similarity: https://www.nltk.org/howto/wordnet.html. We tried other WordNet-based similarities available in nltk (e.g., Wu-Palmer similarity), but the Lin similarity produced the best results in our experiments.

connect coreference event mentions. Connected components from the coreference graph can then be returned to serve as predicted event mention clusters in D.

In order to predict the coreferent antecedent $y_i \in \mathcal{Y}$ for an event mention e_i, we compute the distribution over the possible antecedents in $\mathcal{Y}_i$ for e_i via: $P\left(y_i \middle| e_i, \mathcal{Y}_i\right) = \frac{e^{s(e_i, y_i)}}{\sum_{y' \in \mathcal{Y}(i)} e^{s(e_i, y')}}$ where $s(e_i, e_j)$ is a score function to determine the coreference likelihood between e_i and e_j in D. To this end, we set $s(e_i, \epsilon) = 0$ for all $e_i \in E$. Inspired by (Lee et al., 2017b), we obtain the score function $s(e_i, e_j)$ for e_i and e_j by leveraging their GCN-induced representation vectors r_{e_i} and r_{e_j} via:

$$
\begin{aligned}
&s(e_i, e_j) = s_m(e_i) + s_m(e_j) + s_c(e_i, e_j) + s_a(e_i, e_j) \\
&s_m(e_i) = \boldsymbol{w}_m^\top \texttt{FF}_m(r_{e_i}) \\
&s_c(e_i, e_j) = \boldsymbol{w}_a^\top \texttt{FF}_c([r_{e_i}, r_{e_j}, r_{e_i} \odot r_{e_j}]) \\
&s_a(e_i, e_j) = r_{e_i}^\top \boldsymbol{W}_c r_{e_j}
\end{aligned}
$$

where $\texttt{F}_m$ and $\texttt{FF}_c$ are two-layer feed-forward networks, $\boldsymbol{w}_m^\top$ and $\boldsymbol{w}_a^\top$ are learnable vectors, $\boldsymbol{W}_c$ is a weight matrix, and $\odot$ is the element-wise multiplication. At the inference time, we employ the greedy decoding to predict the antecedent $\hat{y}_i$ for e_i: $\hat{y}_i = \text{argmax} P(y_i|e_i, \mathcal{Y}_i)$. For training, we use the negative log-likelihood as the loss function in our end-to-end framework: $\mathcal{L}_{pred} = -\sum_{i=0}^{|E|} \log P(y_i^*|e_i, \mathcal{Y}_i)$ ($y_i^* \in \mathcal{Y}_i$ is the golden antecedent for e_i).

3.4 Cluster Consistency Regularization

To further improve representation learning for ECR, we propose to regularize the induced representation vectors of the event mentions in E to explicitly enforce the consistency/similarity between golden and predicted event mention clusters in D. This is based on our motivation that ECR models will perform better if they can produce more similar event mention clusters to the golden ones. As such, for convenience, let $\mathcal{T} = \{T_1, T_2, \ldots, T_{|\mathcal{T}|}\}$ and $\mathcal{P} = \{P_1, P_2, \ldots, P_{|\mathcal{P}|}\}$ be the golden and predicted sets of event mentions in E respectively, i.e., $T_i, P_j \subset E$, and $T_1 \cup T_2 \cup \ldots \cup T_{|\mathcal{T}|} = P_1 \cup P_2 \cup \ldots \cup P_{|\mathcal{P}|} = E$. Also, for each cluster C in $\mathcal{T}$ or $\mathcal{P}$, we compute a centroid vector r_C for it by averaging the representation vectors of the event mention members: $r_C = \text{average}_{e \in C}(r_e)$. This leads to the centroid vectors $\{r_{T_1}, r_{T_2}, \ldots, r_{T_{|\mathcal{T}|}}\}$ and $\{r_{P_1}, r_{P_2}, \ldots, r_{P_{|\mathcal{P}|}}\}$ for $\mathcal{T}$ and $\mathcal{P}$ respectively. We propose the following regularization terms for cluster consistency:

Intra-cluster Consistency: This constraint concerns the inner information of each cluster, characterizing the structure of each individual event mention in its golden and predicted clusters in $\mathcal{T}$ and $\mathcal{P}$. In particular, for each event mention $e_i \in E$, we expect its distances to the centroid vectors of the corresponding golden and predicted clusters T'_i and P'_i in $\mathcal{T}$ and $\mathcal{P}$ (respectively) to be similar, i.e., $T'_i \in \mathcal{T}, P'_i \in \mathcal{P}, e_i \in T'_i, e_i \in P'_i$. As such, we compute the distances between the representation vector r_{e_i} of e_i to the centroid vectors $r_{T'_i}$ and $r_{P'_i}$ via the Euclidean distances $\|r_{e_i} - r_{T'_i}\|_2^2$ and $\|r_{e_i} - r_{P'_i}\|_2^2$. Afterward, the differences $\mathcal{L}_{inner}$ between the two distances for golden and predicted clusters are aggregated over all event mentions and added into the overall loss function for minimization: $\mathcal{L}_{inner} = \sum_{i=1}^{|E|} \left| \|r_{e_i} - r_{T'_i}\|_2^2 - \|r_{e_i} - r_{P'_i}\|_2^2 \right|$.

Inter-cluster Consistency: In this constraint, we expect that the structure among the clusters T_i in the golden set $\mathcal{T}$ is consistent with those for the predicted event cluster set $\mathcal{P}$ (i.e., inter-cluster regulation). To implement this idea, we encode the structure of the clusters in a set via the average of the pairwise distances between the centroid vectors of the clusters. In particular, the inter-cluster structure scores for the golden and predicted clusters in $\mathcal{T}$ and $\mathcal{P}$ are computed via: $s_{\mathcal{T}} = \frac{2}{|\mathcal{T}|(|\mathcal{T}|-1)} \sum_{i=1}^{|\mathcal{T}|} \sum_{j=i+1}^{|\mathcal{T}|} \|r_{T_i} - r_{T_j}\|_2^2$, and $s_{\mathcal{P}} = \frac{2}{|\mathcal{P}|(|\mathcal{P}|-1)} \sum_{i=1}^{|\mathcal{P}|} \sum_{j=i+1}^{|\mathcal{P}|} \|r_{P_i} - r_{P_j}\|_2^2$. The difference between the structure scores for golden and predicted clusters $\mathcal{T}$ and $\mathcal{P}$ is then included into the overall loss function for minimization: $\mathcal{L}_{inter} = |s_{\mathcal{T}} - s_{\mathcal{P}}|$.

Inter-set Similarity: This constraint aims to directly promote the similarity between the golden clusters in $\mathcal{T}$ and the predicted clusters in $\mathcal{P}$. As such, for the golden and predicted cluster sets $\mathcal{T}$ and $\mathcal{P}$, we first obtain the overall centroid vectors $u_{\mathcal{T}}$ and $u_{\mathcal{P}}$ (respectively) by averaging the centroid vectors of their member clusters: $u_{\mathcal{T}} = \text{average}_{T \in \mathcal{T}}(r_T)$ and $u_{\mathcal{P}} = \text{average}_{P \in \mathcal{P}}(r_P)$. The Euclidean distance $\mathcal{L}_{sim}$ is then integrated into the overall loss for minimization: $\mathcal{L}_{sim} = \|u_{\mathcal{T}} - u_{\mathcal{P}}\|_2^2$. Note that $\mathcal{L}_{inner}$, $\mathcal{L}_{inter}$, and $\mathcal{L}_{sim}$ will be zero if the predicted clusters in $\mathcal{P}$ are the same as those in the golden clusters in $\mathcal{T}$.

To summarize, the overall loss function $\mathcal{L}$ to train our ECR model in this work is: $\mathcal{L} = \mathcal{L}_{pred} + \alpha_{inner}\mathcal{L}_{inner} + \alpha_{inter}\mathcal{L}_{inter} + \alpha_{sim}\mathcal{L}_{sim}$ with α_{inner}, α_{inter}, and α_{sim} as the trade-off parameters.

4 Experiments

4.1 Dataset & Hyperparameters

Following prior work (Choubey and Huang, 2018), we train our ECR models on the KBP 2015 dataset (Mitamura et al., 2015) and evaluate the models on the KBP 2016 and KBP 2017 datasets for ECR (Mitamura et al., 2016, 2017). In particular, the KBP 2015 dataset includes 360 annotated documents for ECR (181 documents from discussion forum and 179 documents from news articles). We use the same 310 documents from KBP 2015 as in (Choubey and Huang, 2018) for the training data and the remaining 50 documents for the development data. Also, similar to (Choubey and Huang, 2018), the news articles in KBP 2016 (85 documents) and KBP 2017 (83 documents) are leveraged for test datasets. To ensure a fair comparison, we use the predicted event mentions provided by (Choubey and Huang, 2018) in all the datasets. Finally, we report the ECR performance based on the official KBP 2017 scorer (version 1.8)[3]. The scorer employs four coreference scoring measures, including MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), CEAF-e (Luo, 2005), BLANC (Lee et al., 2012b), and the unweighted average of their F1 scores (AVG$_{F1}$).

Hyper-parameters for the models are fine-tuned by the AVG$_{F1}$ scores over development data. The selected values from the tuning process include: $1e$-5 for the learning rate of the Adam optimizer (selected from $[1e$-5, $2e$-5, $3e$-5, $4e$-5, $5e$-5$]$); 8 for the mini-batch size (selected from $[8, 16, 32, 64]$); 128 hidden units for all the feed-forward network and GCN layers (selected from $[64, 128, 256, 512]$); 2 layers for the GCN model, $G = 2$ (selected from $[1, 2, 3, 4]$), and $\alpha_{inner} = 0.1$, $\alpha_{inter} = 0.1$, and $\alpha_{sim} = 0.1$ for the trade-off parameters in the overall loss function $\mathcal{L}$ (selected from $[0.1, 0, 2, \ldots, 0.9]$). Finally, we use the BERT$_{base}$ model (of 768 dimensions) for the pre-trained word embeddings (updated during the training).

4.2 Performance Evaluation

We compare the proposed model for ECR with document structures and cluster consistency regularization (called StructECR) with prior work ECR models in the same evaluation setting, including the joint model between ECR and event detection (Lu and Ng, 2017), the integer linear programming approach in (Choubey and Huang, 2018), and the discourse structure profiling model in (Choubey et al., 2020) (also the model with the best reported performance in KBP datasets). In addition, we examine the following baselines of StructECR to highlight the benefits of the proposed components:

E2E-Only: This variant implements the end-to-end resolution model described in Section 3.3 where all event mentions in a document are resolved simultaneously in a single process. However, different from our full model StructECR, E2E-Only does not include the document structure component with GCN for representation learning, i.e., it directly uses the initial representation vectors v_i (induced from BERT) for the event mentions in the computation of the distribution $P(y_i|e_i, \mathcal{Y}_i)$. Also, the cluster consistency regularization in Section 3.4 is also not included in this model.

Pairwise: This model is similar to E2E-Only in that it does not applies the document structures and regularization terms in StructECR. In addition, instead of simultaneously resolving event mentions in documents, Pairwise predicts the coreference for every pair of event mentions separately. In particular, the representation vectors v_{e_i} and v_{e_j} for two event mentions e_i and e_j (included from BERT) are combined via $[v_{e_i}, v_{e_j}, v_{e_i} \odot v_{e_j}]$. This vector is then sent into a feed-forward network to produce a distribution over possible coreference labels between e_i and e_j (i.e., two labels for being coreferent or not). The coreference labels for every pair of event mentions are then gathered in a coreference graphs among event mentions; the connected components will be returned for the event clusters.

Table 1 reports the performance of the ECR models on the KBP 2016 and KBP 2017 datasets. As can be seen from the table, E2E-Only performs comparably or better than prior state-of-the-art models for ECR, e.g., (Choubey and Huang, 2018) and (Choubey et al., 2020), that employ extensive feature engineering. In addition, the better performance of E2E-Only over Pairwise (for both KBP 2016 and KBP 2017) illustrates the benefits of end-to-end coreference resolution for event mentions in documents. Most importantly, the proposed model StructECR significantly outperforms all the baseline models for which the performance improvement over E2E-Only is 1.94% and 1.26% (i.e., AVG$_{F1}$ scores) over the KBP 2016 and KBP 2017 datasets respectively. This clearly demonstrates the benefits of the proposed ECR model with rich

[3] https://github.com/hunterhector/EvmEval

	KBP 2016					KBP 2017				
Model	B^3	$CEAF_e$	MUC	BLANC	AVG$_{F1}$	B^3	$CEAF_e$	MUC	BLANC	AVG$_{F1}$
(Lu and Ng, 2017)	50.16	48.59	32.41	32.72	40.97	-	-	-	-	-
(Choubey and Huang, 2018)	51.67	49.10	34.08	34.08	42.23	50.35	48.61	37.24	31.94	42.04
(Choubey et al., 2020)	52.78	49.70	34.62	34.49	42.90	51.68	50.57	37.8	33.39	43.36
Pairwise	52.16	49.84	30.79	32.21	41.25	50.97	48.80	36.92	31.86	42.14
E2E-Only	50.89	50.43	36.05	33.93	42.83	51.60	52.03	38.53	33.02	43.80
StructECR	52.77	52.29	38.37	35.66	**44.77**	51.93	52.82	40.73	34.75	**45.06**

Table 1: Models' performance on the KBP 2016 and KBP 2017 datasets. The performance improvement of StructECR over E2E-Only is significant with $p < 0.01$.

Model	B^3	$CEAF_e$	MUC	BLANC	AVG$_{F1}$
StructECR (full)	76.86	69.99	66.40	69.02	**70.57**
StructECR - a_{ij}^{sent}	75.37	69.73	62.42	69.49	69.25
StructECR - a_{ij}^{coref}	75.07	69.74	62.97	69.67	69.36
StructECR - a_{ij}^{span}	75.32	70.32	63.44	66.97	69.01
StructECR - a_{ij}^{dep}	74.66	69.76	62.72	69.14	69.07
StructECR - a_{ij}^{struct}	75.44	69.53	61.82	71.48	69.57
StructECR - Entity Nodes	74.67	69.71	63.01	67.35	68.69
StructECR - GraphCombine	75.41	69.74	62.38	68.90	69.11
StructECR - Doc Structures	74.15	66.78	60.24	66.32	66.87
StructECR - $\mathcal{L}_{inner}$	75.09	68.44	62.25	68.01	68.45
StructECR - $\mathcal{L}_{inter}$	74.80	67.98	61.92	67.71	68.10
StructECR - $\mathcal{L}_{sim}$	75.13	68.12	62.03	68.95	68.56
StructECR - Regularization	74.46	67.55	60.74	68.28	67.76

Table 2: Performance on the KBP 2015 dev set.

	KBP 2016		KBP 2017	
	NW → DF	DF → NW	NW → DF	DF → NW
Pairwise	60.51	58.11	59.22	59.10
E2E-Only	65.82	62.01	62.56	62.52
StructECR	**68.19**	**65.83**	**65.19**	**65.12**

Table 3: Cross-domain performance (AVG$_{F1}$). NW and DF represent news articles and discussion forum documents respectively. X → Y implies models trained on domain X and tested on domain Y.

document structures and cluster consistency regularization for representation learning.

4.3 Ablation Study

Two major components in the proposed model StructECR involve the document structures and the cluster consistency regularization. This section performs an ablation study to reveal the contribution of such components for the full model. First, for the document structures, we examine the following ablated models: (i) "**StructECR - x**": where **x** is one of the five interaction scores used to compute the unified score a_{ij} for $\mathcal{G}$ (i.e., a_{ij}^{sent}, a_{ij}^{coref}, a_{ij}^{span}, a_{ij}^{dep}, and a_{ij}^{struct}). For example, "**StructECR -** a_{ij}^{span}" implies a variant of StructECR where the span-based interaction score a_{ij}^{span} is not included in the compuation of the overall score a_{ij}; (ii) "**StructECR - Entity Nodes**: this model excludes the entity mention nodes from the interaction graph $\mathcal{G}$ in StructECR (i.e., $\mathcal{N} = D \cup E$ only); (iii) "**StructECR - GraphCombine**": instead of unifying the five interaction scores in d_{ij} into an overall score a_{ij} in Equation 1, this model considers each of the five generated interaction scores as forming a separate interaction graph, thus producing six different graphs. The GCN model is then applied over those five graphs (using the same initial representation vectors v_i for the nodes n_i in $\mathcal{N}$). The outputs of the GCN model for the same node n_i (with different graphs) are then concatenated to compute the final representation vector h_i for n_i; and (iv) **StructECR - Doc Structures**: this model removes the GCN model from StructECR. As such, the interaction graph $\mathcal{G}$ is not used and the GCN-induced representation vectors h_i are replaced by the initial BERT-induced representation vectors v_i in the computation for end-to-end resolution and consistency regularization.

Second, for the cluster consistency regularization, we evaluate the following ablated models for StructECR: (v) **StructECR - y** (**y** $\in \{\mathcal{L}_{inner}, \mathcal{L}_{inter}, \mathcal{L}_{sim}\}$): these models exclude one of the regularization terms for the consistency between golden and predicted clusters from the overall loss function $\mathcal{L}$; and (vi) **StructECR - Regularization**: this model completely ignores the consistency regularization component from StructECR.

Table 2 shows the performance of the models on the development data of the KBP 2015 dataset. As can be seen, the elimination of any component from StructECR would significantly hurt the performance, thus clearly demonstrating the benefits of the designed document structures and cluster consistency regularization in StructECR.

4.4 Cross-domain Evaluation

To further demonstrate the benefits for the proposed model StructECR, we evaluate StructECR and the baseline models Pairwise and E2E-Only in the cross-domain setting. In this setting, we aim to train the models on one domain (the source domain) and evaluate them on another domain (the target domain). We leverage the KBP 2016 and KBP 2017 datasets for this experiment. In particular, KBP 2016 annotates ECR data for 85 newswire and 84 discussion forum documents (i.e., two domains/genres) while KBP 2017 provides annotated data for ECR on 83 news articles and 84 discussion forum documents. As such, for each dataset, we consider two setups where documents in one domain (i.e., newswire or discussion forum) are used for the source domain, leaving documents in the other domain for the target domain data. We use the same hyper-parameters that are tuned on the development set of KBP 2015 for the models in this experiment. Table 3 presents the performance of the models. It is clear from the table that StructECR are significantly and substantially better than the baseline models ($p < 0.01$) over different datasets and settings for the source and target domains, thereby confirming the domain generalization advantages of StructECR for ECR.

5 Conclusion

We present a novel end-to-end coreference resolution framework for event mentions based on deep learning. The novelty in our model is twofold. First, document structures are introduced to explicitly capture relevant objects and their interactions in documents to aid representation learning. Second, several regularization techniques are proposed to exploit the consistencies between human-provided and machine-generated clusters of event mentions in documents. We perform extensive experiments on two benchmark datasets for ECR to demonstrate the advantages of the proposed model. In the future, we plan to extend our models to related problems in information extraction, e.g., event extraction.

Acknowledgments

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112. 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

Cosmin Adrian Bejan and Sanda Harabagiu. 2014. Unsupervised event coreference resolution. In *Computational Linguistics (CL).*

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*.

Jun Araki and Teruko Mitamura. 2015. Joint event trigger identification and event coreference resolution with structured perceptron. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the International Conference on Computational Linguistics (COLING).*

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL).*

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL).*

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2020. Streamlining cross-document coreference resolution: Evaluation and modeling. *arXiv preprint arXiv:2009.11032*.

Chen Chen and Vincent Ng. 2016. Joint inference over a lightly supervised information extraction pipeline: Towards event coreference resolution for resource-scarce languages. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI).*

Zheng Chen and Heng Ji. 2009. Graph-based event coreference resolution. In *Proceedings of the Workshop on Graph-based Methods for Natural Language Processing*.

Zheng Chen, Heng Ji, and Robert Haralick. 2009. A pairwise event coreference model, feature impact and evaluation for event coreference resolution. In *Proceedings of the Workshop on Events in Emerging Text Types*.

Prafulla Kumar Choubey and Ruihong Huang. 2017. Event coreference resolution by iteratively unfolding inter-dependencies among events. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Prafulla Kumar Choubey and Ruihong Huang. 2018. Improving event coreference resolution by modeling correlations between event coreference chains and document topic structures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Prafulla Kumar Choubey, Aaron Lee, Ruihong Huang, and Lu Wang. 2020. Discourse as a function of event: Profiling discourse structure in news articles around the main event. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Agata Cybulska and Piek T. J. M. Vossen. 2015. ”bag of events” approach to event coreference resolution. supervised classification of event templates. In *Int. J. Comput. Linguistics Appl.*

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yin Jou Huang, Jing Lu, Sadao Kurohashi, and Vincent Ng. 2019. Improving event coreference resolution by learning argument compatibility from unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Mandar Joshi, Danqi Chen, Y. Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019a. Spanbert: Improving pre-training by representing and predicting spans. In *Transactions of the Association for Computational Linguistics (TACL)*.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019b. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kian Kenyon-Dean, Jackie Chi Kit Cheung, and Doina Precup. 2018. Resolving event coreference with supervised representation learning and clustering-oriented regularization. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM)*.

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

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012a. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012b. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017a. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017b. End-to-end neural coreference resolution. In *Proceedings of the 2017 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 51th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Dekang Lin et al. 1998. An information-theoretic definition of similarity. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Zhengzhong Liu, Jun Araki, Eduard Hovy, and Teruko Mitamura. 2014. Supervised within-document event coreference using information propagation. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Jing Lu and Vincent Ng. 2017. Joint learning for event coreference resolution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jing Lu, Deepak Venugopal, Vibhav Gogate, and Vincent Ng. 2016. Joint inference for event coreference resolution. In *Transactions of the Association for Computational Linguistics (TACL)*.

Yaojie Lu, Hongyu Lin, Jialong Tang, Xianpei Han, and Le Sun. 2020. End-to-end neural event coreference resolution. In *CoRR abs/2009.08153*.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Teruko Mitamura, Zhengzhong Liu, and Eduard H. Hovy. 2015. Overview of TAC-KBP 2015 event nugget track. In *Proceedings of the Text Analysis Conference (TAC)*.

Teruko Mitamura, Zhengzhong Liu, and Eduard H. Hovy. 2016. Overview of TAC-KBP 2016 event nugget track. In *Proceedings of the Text Analysis Conference (TAC)*.

Teruko Mitamura, Zhengzhong Liu, and Eduard H. Hovy. 2017. Events detection, coreference and sequencing: What's next? overview of the TAC KBP 2017 event track. In *Proceedings of the Text Analysis Conference (TAC)*.

Vincent Ng. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Thien Huu Nguyen, , Adam Meyers, and Ralph Grishman. 2016. New york university 2016 system for kbp event nugget: A deep learning approach. In *Proceedings of the Text Analysis Conference (TAC)*.

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)*.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multipass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020a. Exploiting the syntax-model consistency for neural relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Amir Pouran Ben Veyseh, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020b. 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)*.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6)*.

Bishan Yang, Claire Cardie, and Peter Frazier. 2015. A hierarchical distance-dependent Bayesian model for event coreference resolution. In *Transactions of the Association for Computational Linguistics (TACL)*.