End-to-End Neural Event Coreference Resolution

Yaojie Lu[1,3], **Hongyu Lin**[1], **Jialong Tang**[1,3], **Xianpei Han**[1,2] , **Le Sun**[1,2]
[1]Chinese Information Processing Laboratory [2]State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences, Beijing, China
[3]University of Chinese Academy of Sciences, Beijing, China
{yaojie2017,hongyu2016,jialong2019,xianpei,sunle}@iscas.ac.cn

Abstract

Traditional event coreference systems usually rely on pipeline framework and hand-crafted features, which often face error propagation problem and have poor generalization ability. In this paper, we propose an **E**nd-to-**E**nd **E**vent **C**oreference approach – E^{3}C neural network, which can jointly model event detection and event coreference resolution tasks, and learn to extract features from raw text automatically. Furthermore, because event mentions are highly diversified and event coreference is intricately governed by long-distance, semantic-dependent decisions, a type-guided event coreference mechanism is further proposed in our E^{3}C neural network. Experiments show that our method achieves new state-of-the-art performance on two standard datasets.

1 Introduction

Event coreference resolution aims to identify which event mentions in a document refer to the same event (Ahn, 2006; Hovy et al., 2013). For example, the two event mentions in Figure 1, *departing* and *leave*, refer to the same *EndPosition* event of Nokia's CEO.

Traditional event coreference resolution methods usually rely on a series of upstream components (Lu and Ng, 2018), such as entity recognition and event detection. Such a pipeline framework, unfortunately, often suffers from the error propagation problem. For instance, the best event detection system in KBP 2017 only achieved 56 F1 (Jiang et al., 2017), and it will undoubtedly limit the performance of the follow-up event coreference task (35 Avg F1 on KBP 2017). Furthermore, most previous approaches use hand-crafted features (Chen et al., 2011; Lu and Ng, 2017a), which heavily depend on other NLP components (e.g., POS tagging, NER, syntactic parsing, etc.) and thus are hard to generalize to new languages/domains/datasets.

Huge Payday for Nokia's [departing]$_{\text{EndPosition}}$ CEO ... *Nokia's CEO prepares to* [leave]$_{\text{EndPosition}}$ *the company and* [rejoin]$_{\text{StartPosition}}$ *Microsoft ... It is an expensive* [goodbye]$_{\text{EndPosition}}$ *for the executive, ...*

Figure 1: An example of event coreference resolution, which contains two coreferential chains: An *EndPosition* event chain {*departing*, *leave*, *goodbye*} and a *StartPosition* chain {*rejoin*}.

In this paper, we propose an **E**nd-to-**E**nd **E**vent **C**oreference method – E^{3}C neural network, which can predict event chains from a raw text in an end-to-end manner. For example, taking the raw text in Figure 1 as input, E^{3}C will directly output two event coreference chains, {*departing*, *leave*, *goodbye*} and {*rejoin*}. By jointly modeling event detection and event coreference, E^{3}C neural network does not require any prior components, and the representations/pieces of evidence between different tasks and different decisions can be shared and reinforced. Besides, E^{3}C are learned in an end-to-end manner, which can inherently resolve the error propagation problem.

End-to-end event coreference, however, is challenging due to the mention diversity and the long-distance coreference. *First, event mentions are highly diversified* (Humphreys et al., 1997; Chen and Ji, 2009), which may be a variety of syntactic objects, including nouns, verbs, and even adjectives. For example, an *EndPosition* event can be triggered by *departing*(noun), *leave*(verb), *goodbye*(noun) and *former*(adj). By contrast, mentions in entity coreference are mostly noun phrases (Lu and Ng, 2018). *Second, coreferential event mentions commonly appear over long-distance sentences, therefore event coreference is intricately governed by long-distance, semantic-dependent decisions* (Choubey and Huang, 2018; Goyal et al., 2013; Peng et al., 2016). For example, in Figure

1 the closest antecedent[1] of the mention *goodbye* – *leave*, is far from it. To resolve the coreference between these two distant, diverse event mentions, a system can only rely on their semantic meanings, i.e., they both describe the same *EndPosition* event(the departing of Nokia's CEO) but from different perspectives. By contrast, most of entity mentions' closest antecedents are in the same or immediately preceding sentence (Choubey and Huang, 2018), which can be resolved more easily using local and syntactic clues.

To resolve the mention diversity problem and the long-distance coreference problem, this paper further proposes a type-guided mechanism into our E^{3}C neural network. This mechanism bridges distant, diverse event mentions by exploiting event type information in three folds: 1) **type-informed antecedent network** which enables E^{3}C to capture more semantic information of event mentions by predicting coreferential scores and type scores simultaneously; 2) **type-refined mention representation** which enhances mention representation with type information, therefore even lexically dissimilar mentions can be bridged together, such as the two diverse *EndPosition* mentions *goodbye* and *departing*; 3) **type-guided decoding algorithm** which can exploit global type consistency for more accurate event chains.

The main contributions of this paper are:

1. We propose an end-to-end neural network for event coreference resolution - E^{3}C neural network. E^{3}C can jointly model event detection and event coreference, and learn to automatically extract features from raw text. To the best of our knowledge, this is the first end-to-end neural event coreference model that can achieve state-of-the-art performance.

2. We design a type-guided mechanism for event coreference, which can effectively resolve the mention diversity problem and the long-distance coreference problem in event coreference resolution.

3. We conduct experiments on two standard datasets: KBP 2016 and KBP 2017, which show that E^{3}C achieves new state-of-the-art performance. And additional ablation experiments verify the effectiveness of the proposed type-guided mechanism.

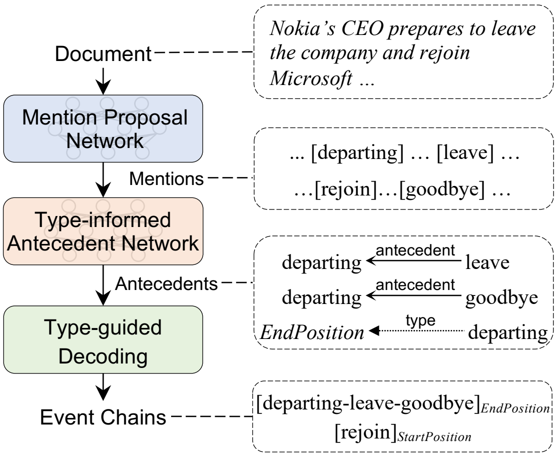
Document
Nokia's CEO prepares to leave the company and rejoin Microsoft ...
Mention Proposal Network
Mentions
... [departing] ... [leave] ...
...[rejoin]...[goodbye] ...
Type-informed Antecedent Network
Antecedents
departing
antecedent
leave
departing
antecedent
goodbye
EndPosition
type
departing
Type-guided Decoding
Event Chains
[departing-leave-goodbye]*EndPosition*
[rejoin]*StartPosition*

Figure 2: The framework of our E^{3}C neural network.

2 E^{3}C: End-to-end Neural Event Coreference Resolution

Given a document $D = \{w_1, ..., w_n\}$, an end-to-end event coreference system needs to: 1) detect event mentions $\{m_1, ., m_l\}$ (event detection); 2) predict all coreference chains $\{ev_*\}$ (event coreference resolution). For example, in Figure 1, the mentions are {*departing*, ..., *goodbye*} and two coreference chains will be predicted: {*departing*, *leave*, *goodbye*}, and {*rejoin*}.

To this end, our E^{3}C method first detects mentions candidates via a mention proposal network, then identifies all mentions' antecedents via an antecedent prediction network. To resolve the mention diversity problem and the long-distance coreference problem, a type-guided event coreference mechanism is designed for E^{3}C. Figure 2 shows the framework of our method. All components in E^{3}C are differentiable and can be trained in an end-to-end manner. In the following, we describe them in detail.

2.1 Proposing Mention Candidates via Mention Proposal Network

The mention proposal network detects all event mentions in a document, e.g., identifying {*departing*, ..., *rejoin*} as event mentions in Figure 1. Because event mentions are highly diversified expressions (e.g., *goodbye*, *former* and *leave* for *EndPosition*), we first capture the semantic information of all tokens via a contextualized representation layer, then identify mention candidates via a mention proposal layer. The details are as follows.

Contextualized Word Representation Layer. To capture the semantic information for proposing

[1] In this paper, antecedents are coreferential mentions that appear earlier in the document.

event mentions, we learn a contextualized representation for each token. Concretely, we first obtain a task-independent representation for each token based on pre-trained BERT embeddings (Devlin et al., 2019). Following Tenney et al. (2019), a token w_i's representation $\mathbf{h}_i \in \mathbb{R}^d$ is pooled across different BERT layers using scalar mixing (Peters et al., 2018) as $\mathbf{h}_i = \gamma \sum_{j=1}^{L} \alpha_j \mathbf{x}_i^{(j)}$, where $\mathbf{x}_i^{(j)}$ is the embedding of token i from BERT layer j, d is size of bert embedding, α_j is softmax-normalized weights, and γ is a scalar parameter.

Because event arguments can provide critical evidence (Bejan and Harabagiu, 2010; Lee et al., 2012; McConky et al., 2012; Cybulska and Vossen, 2013), we further obtain an event-specific token representation by distilling argument information from raw text implicitly. Specifically, we design a mask attention strategy (Dong et al., 2019). Given task-independent token representations $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n\}$, our attention mechanism first models the relevance between tokens via a scaled dot-product attention (Vaswani et al., 2017) without linear projection, and then computes the final contextualized word representations $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_n\}$ as:

$$\begin{aligned} \mathbf{C} &= \text{softmax}\left(\frac{\mathbf{H}\mathbf{H}^T}{\sqrt{d}} + \mathbf{M}\right)\mathbf{H} \\ \mathbf{M}_{ij} &= \begin{cases} 0, & |i-j| < c \\ -\infty & \text{otherwise} \end{cases} \end{aligned} \tag{1}$$

where c is the size of local window (this paper focuses on the local context since arguments empirically appear around event mentions[2], and we set $c = 10$ in this paper), and $\sqrt{d}$ is the scaling factor.

Mention Proposal Layer. Given the token representations, the mention proposal layer assigns a mention score to each span – $s_m(i)$, which indicates the likelihood for span i being an event mention. For example, in Figure 1 the mention proposal layer will assign spans $\{$*departing*, *leave*, *rejoin*, ...$\}$ with high $s_m(i)$ scores because they are highly likely to be event mentions, and assign spans $\{$*prepares to*, *company*, ...$\}$ with low $s_m(i)$ scores because they are unlikely to be event mentions.

Given all spans within a restricted length[3] in a document, the mention proposal layer represents each span i as $\mathbf{g}_i = \hat{\mathbf{c}}_i$, where $\hat{\mathbf{c}}_i$ is the soft head attention-based aggregation of all token representations in span i (Lee et al., 2016). Given $\mathbf{g}_i$, the mention score $s_m(i)$ is computed via standard feed-forward neural networks:

$$s_m(i) = \text{FFNN}_m(\mathbf{g}_i) \tag{2}$$

Finally, we rank all spans according to their mention scores (Lee et al., 2017), and only retain top-l mentions[4] $\{m_1, m_2, ..., m_l\}$ as event mention candidates for computation efficiency.

2.2 Predicting Antecedent via Type-informed Antecedent Network

Given an event mention, the type-informed antecedent network predicts its antecedents, and the antecedent predictions can be used as local pair-wise coreference decisions. For example, our method will predict the antecedent of *leave* as *departing* in Figure 1 and $\langle$departing, leave$\rangle$ can be used as a pair-wise coreference decision.

For each mention m_i in $\{m_1, ..., m_l\}$, the type-guided antecedent network produces two kinds of scores simultaneously: 1) $s(i, j)$ – the score for mention m_j being antecedent of m_i, where m_j must appear before m_i in the document; 2) $s(i, t_k)$ – the score for mention m_i's type being t_k.

Antecedent Score. Given a mention m_i, antecedent network computes an antecedent score $s(i, j)$ for each mention pair $\langle m_i, m_j \rangle$:

$$s(i, j) = s_m(i) + s_m(j) + s_a(i, j) \tag{3}$$

where $j < i$, $s_m(i)$ and $s_m(j)$ are the mention scores described in §2.1; $s_a(i, j)$ measures the semantic similarity between m_i and m_j, computed via a standard feed-forward neural network:

$$s_a(i, j) = \text{FFNN}_a([\mathbf{g}_i, \mathbf{g}_j, \mathbf{g}_i \circ \mathbf{g}_i, \Phi(i, j)]) \tag{4}$$

where $\mathbf{g}_i \circ \mathbf{g}_i$ is the element-wise similarity of each mention pair $\langle m_i, m_j \rangle$, and $\Phi(i, j)$ is the distance encoding between two mentions.

Event Type Score. As described in §1, event coreference is intricately governed by long-distance, semantic-dependent decisions. To address this issue, this paper exploits event type information for better event coreference resolution. Specifically, besides antecedent prediction for each mention, we further predict its event type so that: 1) the neural network will be guided to capture more semantic information about event mentions (Durrett and Klein, 2014); 2) the type information

[2] In KBP 2017 training set, about 90% of arguments appear in the ±10-word window of their trigger word.

[3] This paper restricts span length to 1, which can cover 96.6% mentions in KBP 2017 training set. For this case, the attented span representation $\mathbf{g}_i$ is equivalent to $\mathbf{c}_i$.

[4] In this paper, $l = 0.1\times$ document length.

ensures the global type consistency during coreference resolution, i.e., mentions in the same coreference chain will have the same event type.

Specifically, we first embed all event types $\mathcal{T} = \{t_1, ..., t_t\}$ via a hierarchical embedding algorithm. The embedding of t_k is $\mathbf{g}_{t_k} = \mathbf{W}_e \cdot [\mathbf{e}_{event}, \mathbf{e}_{type}(t_k)]$, where $\mathbf{e}_{event}$ is shared by all event types, $\mathbf{e}_{type}(t_k)$ indicates embedding of t_k, and $\mathbf{W}_e$ is a mapping matrix. The dimension of $\mathbf{g}_{t_k}$ is the same as mention embedding $\mathbf{g}_i$.

Then the type scores $s(i, t_k)$ are computed via the same scoring function for antecedent prediction:

$$\begin{gathered} s_m(t_k) = \text{FFNN}_m(\mathbf{g}_{t_k}) \\ s_a(i, t_k) = \text{FFNN}_a([\mathbf{g}_i, \mathbf{g}_{t_k}, \mathbf{g}_i \circ \mathbf{g}_{t_k}, \Phi(i, t_k)]) \\ s(i, t_k) = s_m(i) + s_m(t_k) + s_a(i, t_k) \end{gathered} \tag{5}$$

where the distance for $\Phi(i, t_k)$ is zero in type scores computation.

For non-mention spans, we add a dummy antecedent ε and assign the antecedents of all non-mention spans to ε, e.g., *company* and *prepares to* in Figure 1. We fix the score $s(i, \varepsilon)$ to 0, and identify a span i as non-mention span if all its antecedent scores $s(i, j) \leq 0$ and all type scores $s(i, t_k) \leq 0$.

In this way, we obtain the antecedent scores and the type scores for each mention via our type-informed antecedent network.

2.3 Enhancing Mention Representation via Type-based Refining

In this section, we describe how to further refine a mentions representation using its type information, so it can capture more semantic information for event coreference resolution. For example, although *goodbye* and *departing* are lexically dissimilar, we can still capture their semantic similarity by further encoding their event type information, i.e., both of them have the same event type – *End-Position*.

To refine mention representation, we first define a probability distribution $Q(t_k)$ over all event types $\mathcal{T}$ and $\{\varepsilon\}$ for each mention span m_i:

$$Q(t_k) = \frac{e^{s(i,t_k)}}{\sum_{t'_k \in \mathcal{T} \bigcup \{\varepsilon\}} e^{s(i,t'_k)}} \tag{6}$$

where $s(i, t_k)$ is the type score. We then obtain an expected event type representation $\tilde{\mathbf{g}}_i$ for each span m_i using the type distribution $Q(t_k)$ as:

$$\tilde{\mathbf{g}}_i = \sum_{t'_k \in \mathcal{T}} Q(t_k = t'_k) \cdot \mathbf{g}_{t'_k} + Q(t_k = \varepsilon) \cdot \mathbf{g}_i \tag{7}$$

Then, we obtain a refined span representation $\mathbf{g}'_i$ by combining its expected event type representation $\tilde{\mathbf{g}}_i$ and its original span representation $\mathbf{g}_i$ via a learnable adaptive gate $\mathbf{f}_i$:

$$\begin{gathered} \mathbf{g}'_i = \mathbf{f}_i \circ \mathbf{g}_i + (1 - \mathbf{f}_i) \circ \tilde{\mathbf{g}}_i \\ \mathbf{f}_i = \sigma(\mathbf{W}_f \cdot [\mathbf{g}_i, \tilde{\mathbf{g}}_i]) \end{gathered} \tag{8}$$

where $\mathbf{W}_f$ is a weight matrix.

Finally, the antecedent network will recompute the coreferential antecedent score $s'(i, j)$ and event type score $s'(i, t_k)$ using the refined span representation $\mathbf{g}'_i$.

2.4 Coreference Resolution via Type-guided Decoding

The type-informed antecedent network produces pairwise coreference scores of mention pairs. To form coreference chains, a naive approach is to directly connect all mentions using their highest-scored antecedent. Unfortunately, such a greedy decoding algorithm only considers local pair-wise consistencies, their results may not be globally optimal, e.g., a coreference chain may contain mentions with different event types.

To address this issue, we propose a decoding algorithm, which can ensure the global consistency of a coreference chain through a type-guided mechanism. For example, to resolve the chain $\{$*departing*, *leave*$\}$ in Figure 1, E^{3}C considers both the antecedent score of $\langle$*departing*, *leave*$\rangle$, and the type consistency that both *departing*, *leave* are *End-Position* mentions.

Concretely, given the mentions $\{m_1, ..., m_l\}$ in a document D, E^{3}C constructs the event coreference chains by sequentially identifying the best antecedent of each mention, further consider the type consistency. For each mention m_i, we first find the mention a_i which has the max coreferential score with m_i where m_j appears before m_i:

$$a_i = \arg \max_{m_j, j<i} s(i, j) \tag{9}$$

and then we check the type consistency between $\langle a_i, m_i \rangle$ by comparing their antecedent score $s(i, a_i)$ and the type prediction score of m_i, $s(i, t_i)$. If $s(i, a_i) > s(i, t_i)$, E^{3}C considers m_i and a_i as type consistent and links mention i to a_i; otherwise, when $s(i, a_i) \leq s(i, t_i)$, E^{3}C considers m_i and a_i as type inconsistent and starts a new event chain for m_i with its type t_i.

3 Model Learning

This section describes how to learn E^{3}C neural network in an end-to-end manner. Given a training corpus $\mathcal{D} = \{D_1, ..., D_N\}$ where each instance D_i is a document with its event mention, mention type, and coreference annotations, our objective function contains two parts: $\mathcal{L}_{antecedent}(\boldsymbol{\Theta})$ – the antecedent loss, and $\mathcal{L}_{proposal}(\boldsymbol{\Theta})$ – the mention proposal loss:

$$\mathcal{L}(\boldsymbol{\Theta}) = \mathcal{L}_{antecedent}(\boldsymbol{\Theta}) + \lambda \mathcal{L}_{proposal}(\boldsymbol{\Theta}) \quad (10)$$

where λ is the coefficient of mention proposal loss (we set $\lambda = 1$ in this paper). This paper optimizes $\boldsymbol{\Theta}$ by maximizing $\mathcal{L}(\boldsymbol{\Theta})$ via Adamax (Kingma and Ba, 2015). The two losses are as follows.

Antecedent Loss. It measures whether a mention links to its correct antecedent. For each mention m_i, this paper identifies its gold antecedent set GOLD(i) as shown in Figure 3:

1) For the first mention of an event chain, the gold antecedent is its event type. For example, the gold antecedent set of *departing* is $\{$*EndPosition*$\}$.

2) For remaining mentions in a chain, the gold antecedents are all its coreferential antecedents. For example, the gold antecedent set of *goodbye* is $\{$*departing*, *leave*$\}$.

3) For non-mention spans, the gold antecedent is the dummy antecedent ε. For example, the gold antecedent set of *company* is $\{\varepsilon\}$.

Given GOLD(i) for each m_i in top-l mention set of document D, the antecedent loss function is a margin log-likelihood function:

$$\mathcal{L}(\boldsymbol{\Theta})_{antecedent} = \log \prod_{i=1}^{l} \sum_{\hat{y} \in \mathcal{Y}(i) \cap \text{GOLD}(i)} P(\hat{y}|D)$$
$$P(y_i|D) = \frac{\exp(s(i, y_i))}{\sum_{y' \in \mathcal{Y}(i)} \exp(s(i, y'))} \quad (11)$$

where $\mathcal{Y}(i)$ is the valid antecedent set for m_i.

Mention Proposal Loss. It measures whether our model can accurately identify event mentions. Specifically, the mention proposal loss uses the binary cross-entropy loss function of the mention proposal network:

$$\mathcal{L}(\boldsymbol{\Theta})_{proposal} = \sum_{i=1}^{n} y_i \log \sigma(s_m(i)) + (1 - y_i) \log(1 - \sigma(s_m(i))) \quad (12)$$

where σ is the sigmoid function, $y_i = 1$ indicates span i is an event mention, otherwise $y_i = 0$.

Annotated Event Chains

EndPosition – *departing* – *leave* – *goodbye*

StartPosition – *rejoin* ε – *company*

Golden Antecedent Sets

GOLD(*departing*) = {*EndPosition*}

GOLD(*rejoin*) = {*StartPosition*}

GOLD(*leave*) = {*departing*}

GOLD(*goodbye*) = {*departing, leave*}

GOLD(*company*) = {ε}

Figure 3: An illustration of gold antecedent sets.

4 Experiments

4.1 Datasets

Following previous studies (Lu and Ng, 2016b, 2017a; Jiang et al., 2017; Huang et al., 2019), we use KBP 2016 and KBP 2017 English datasets for evaluation[5]:

KBP 2016. For KBP 2016, we use the same setup as Lu and Ng (2017a), i.e., 509 documents for training, 139 documents for parameter tuning, and the official KBP 2016 eval set for evaluation.

KBP 2017. Following Huang et al. (2019), we use the English portion of KBP 2015 and 2016 dataset for training, and the KBP 2017 dataset for evaluation. We sample 50 documents from the 2016 evaluation dataset as the validation set.

4.2 Baselines

We compare E^{3}C with the following baselines[6]:

Multi-Pass Sieve (Lu and Ng, 2016a) is an iterative pipeline-based method, which uses both hand-crafted rules and automatic classifiers. We compare two such systems: the Top 1 system in TAC 2016 (Lu and Ng, 2016b) and the Top 1 system in TAC 2017 (Jiang et al., 2017), both of which use additional ensemble strategy for better event detection performance.

[5] There are also other public event coreference datasets: Ontonotes (Pradhan et al., 2007), ECB+ (Bejan and Harabagiu, 2008; Cybulska and Vossen, 2014), ACE (LDC, 2005). Ontonotes and ECB+ are not annotated with event type information therefore is not appropriate for evaluating our end-to-end event coreference model. ACE dataset has strict notion of event identity (Song et al., 2015; Lu and Ng, 2017a), which requires which two event mentions coreferential if and only if "they had the same agent(s), patient(s), time, and location". Because E^{3}C don't perform argument extraction for event coreference, ACE isn't used in this paper. For fair comparsion, we choose the KBP datasets (Ellis et al., 2015, 2016; Getman et al., 2017) so that different systems can be compared in the same settings.

[6] Different from the official type-constraint settings in KBP 2016 and KBP 2017, Choubey and Huang (2018) used relaxed constraints without considering event mention type, so we exclude their system for fair comparison.

		Type-F1	B^3	CEAF$_e$	MUC	BLANC	AVG-F
KBP 2016	Top 1 in TAC 2016 (Lu and Ng, 2016b)	46.99	37.49	34.21	26.37	22.25	30.08
	Mention Ranking (Lu and Ng, 2017b)	46.99	38.64	36.16	26.30	23.59	31.17
	Joint Model (Lu and Ng, 2017a)	49.30	40.90	39.00	27.41	25.00	33.08
	Interact Model$_{\text{BERT}}$	53.65	41.71	38.75	32.17	25.90	34.63
	E^{3}C (*this paper*)	**55.38**	**46.32**	**45.19**	**34.39**	**28.74**	**38.66**
KBP 2017	Top 1 in TAC 2017 (Jiang et al., 2017)	56.19	43.84	39.86	30.63	26.97	35.33
	Interact Model (Huang et al., 2019)	-	42.84	39.01	31.12	24.99	34.49
	+ Transfer (Huang et al., 2019)	-	43.20	40.02	35.66	**32.43**	36.75
	Interact Model$_{\text{BERT}}$	56.54	45.82	44.89	33.61	28.49	38.20
	E^{3}C (*this paper*)	**58.33**	**47.77**	**45.97**	**39.06**	30.60	**40.85**

Table 1: Overall performance on KBP 2016 and KBP 2017 datasets, the results of baselines are adapted from their original papers.

Mention Ranking (Lu and Ng, 2017b) ranks the candidate antecedents of all event mentions and selects the top-ranked antecedent for each mention.

Joint Model (Lu and Ng, 2017a) is a hand-crafted feature-based system that addresses the error propagation problem by jointly learning event trigger detection, event coreference resolution, and event anaphoricity prediction tasks.

Interact Model (Huang et al., 2019) is the state-of-the-art pair-wise method which decides whether two mentions are coreferential using an interactive binary classifier, and then link coreferential mentions to produce final event chains. We also compare with an enhanced model that transfers argument compatibility features from external unlabeled data Interact Model + Transfer. We also reimplement the interact model using BERT as its feature extractor Interact Model$_{\text{BERT}}$, therefore E^{3}C and Interact Model can be compared with the same feature extractors.

4.3 Evaluation Metrics

We use the standard evaluation metrics in KBP evaluation, and compute them using the official evaluation toolkit[7]. We use 4 measures: MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), CEAF$_e$ (Luo, 2005), and BLANC (Recasens and Hovy, 2011). Following previous studies (Lu and Ng, 2017a; Huang et al., 2019), the primary metric AVG-F is the unweighted average of the above four F-scores. We also report the event detection performance using the typed F1-scores as Type-F1.

4.4 Overall Performance

Table 1 shows the overall performance on KBP 2016 and KBP 2017. We can see that:

1. **E^{3}C neural network achieves state-of-the-art performance on both datasets.** Compared with all baselines, E^{3}C gains at least 11.6% and 6.9% AVG-F improvements on KBP 2016 and KBP 2017, respectively. This verifies the effectiveness of the end-to-end framework and the type-guided event coreference mechanism.

2. **By jointly modeling all tasks together and learning all components in an end-to-end manner, E^{3}C neural network significantly outperforms pipeline baselines.** Compared with Interact Model$_{\text{BERT}}$ which uses the same BERT-based feature extractors, E^{3}C still gains 3.2% Type-F1 and 6.9% AVG-F improvements on KBP 2017. This verified the effectiveness of the end-to-end training on reducing the error propagation problem. Besides, by modeling all tasks together, representations and pieces of evidence can be shared and reinforced between different decisions and tasks.

4.5 Detailed Analysis

In this section, we analyze the effects of type-guided mechanism, end-to-end learning, and pre-trained models.

Effect of Type Guided Mechanism. To investigate the effect of type-guided mechanism in E^{3}C, we conduct ablation experiments by ablating type-refined representation (-Type-Refined) and by replacing type-guided decoding with the naive best antecedent decoding (-Type-Guided). Type Rule is a simple heuristic method that regards all event mentions in the same type are coreferential. The results are shown in Table 2. We can see that:

1) Type-guided decoding is effective for event coreference. By considering both type consistency and antecedent score, E^{3}C obtains an 8.1% (3.05) AVG-F improvement over naive decoding.

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

2) Type-refined representation helps resolve the mention diversity problem and the long-distance coreference problem. By incorporating type information into mention representation, E^{3}C obtains a 3.1% (1.24) AVG-F improvement.

Effect of End-to-end Learning. To investigate the effect of end-to-end learning, we conduct experiments on three variations of E^{3}C: E^{3}C$_{\text{Two Stage}}$ which models event mention detection and coreferential antecedent prediction in two independent models but they share span embeddings; E^{3}C$_{\text{w/o Proposal Loss}}$ which removes the mention proposal loss; E^{3}C$_{\text{GoldMention}}$ which uses gold mentions for coreference resolution and type scoring, but the model still needs to predict the type of each mention. Table 3 shows the performances of the three systems, we can find that:

1) One pass paradigm for E^{3}C can effectively share and reinforce the decisions between two tasks. Compared with E^{3}C$_{\text{Two Stage}}$, which has a comparable event detection performance (Type-F1), E^{3}C gains 5.2% AVG-F on the downstream event coreference task.

2) Incorporating mention proposal loss can significantly enhance mention detection performance. By removing mention proposal loss, E^{3}C will loss 2.3% and 4.2% on Type-F1 and AVG-F, respectively. Additionally, the coreference performance can be further significantly improved if golden mentions are used – from 40.89 E^{3}C to 53.72 of E^{3}C$_{\text{GoldMention}}$. This shows that event detection is still a bottleneck for event coreference.

Effect of Pre-trained Models. Pre-trained models are important for neural network-based methods. To investigate their effect on E^{3}C, Table 4 shows the performance of E^{3}C using ELMo (Peters et al., 2018), BERT$_{\text{BASE-Cased}}$, BERT$_{\text{LARGE-Uncased}}$, BERT$_{\text{LARGE-WWM-Uncased}}$ (Devlin et al., 2019), GloVe (Pennington et al., 2014) 300-dimensional word embedding and char embeddings where the contextual layer is BiLSTM. We can find that:

1) Due to the diversity of event mentions, pretrained contextualized embeddings are critical for mention representation. All contextualized embeddings outperform GloVe by a large margin in both event detection and event coreference.

2) E^{3}C can be further improved by employing better pre-trained contextual embeddings. Compared with BERT$_{\text{BASE-Uncased}}$ used in this paper, E^{3}C equipped with BERT$_{\text{LARGE-WWM-Uncased}}$ gains 1.6% Type-F1 and 3.4% AVG-F improvements.

	AVG-F	**Δ**
E^{3}C	40.85	
- Type-Refined	39.61	-1.24
- Type-Guided	37.80	-3.05
Type Rule	31.68	-9.17

Table 2: Ablation results of type-guided mechanism on KBP 2017.

	Type-F1	**AVG-F**
E^{3}C	58.33	40.85
E^{3}C$_{\text{Two Stage}}$	57.63	38.82
E^{3}C$_{\text{w/o Proposal Loss}}$	56.98	39.14
E^{3}C$_{\text{GoldMention}}$	72.73	53.72

Table 3: Performance of different E^{3}C settings on KBP 2017.

E^{3}C	**Type-F1**	**AVG-F**
BERT$_{\text{BASE-Uncased}}$ (*this paper*)	58.33	40.85
GloVe + Char + BiLSTM	52.45	36.43
ELMo	55.24	37.27
BERT$_{\text{BASE-Cased}}$	57.08	39.14
BERT$_{\text{LARGE-Uncased}}$	58.05	40.99
BERT$_{\text{LARGE-WWM-Uncased}}$	**59.29**	**42.23**

Table 4: Performance using different pretrained models for E^{3}C on KBP 2017.

4.6 Discussions

Event Detection Bottleneck. From the above experiments, we find that one main bottleneck of event coreference is event detection. As shown in Table 3, using gold mentions results in a massive improvement on AVG-F, from 40.85 to 53.72. Besides, even if we fix all coreference link errors in predicted event detection results, the growth of AVG-F is still limited, from 40.85 to 42.80. Event detection is challenging because: 1) Event mentions are diversified and ambiguous, detecting them requires a deep understanding of contexts. 2) Some event mentions are multi-tagged[8], i.e., one span triggering multiple events. Because this paper does not consider this issue, it misses some mentions.

Domain Adaptation. We find that domain adaptation is another challenge for event coreference. Table 5 shows the results of our E^{3}C model on different genres of KBP 2017 evaluation dataset: 83 newswire documents – NW, and 84 discussion

[8] 10.18% in KBP 2016 and 8.4% in KBP 2017

	Type-F1	**AVG-F**
NW	59.27	42.39
DF	57.38	39.28

Table 5: Results on subsets of different genres in KBP 2017. NW indicates newswire documents, while DF indicates discussion forum threads.

forum threads – DF. There is a significant performance gap between the two genres, probably because: 1) Different from formal NW documents, DF threads are often informal and lack coherent discourse structures (Choubey and Huang, 2018). 2) Event chains in a discussion forum thread are not only relevant to contents, but also to speaker information and discussion topic. Solving this problem requires a deep understanding of dialogue contexts.

Argument Modeling. In this paper, we exploited the arguments information implicitly via a mask attention strategy, without explicitly extracting argument role. However, we believe event coreference can be further enhanced by modeling argument information more effectively: 1) incorporating explicit argument information can effectively capture semantic information of events for better feature representation (Peng et al., 2016; Choubey and Huang, 2017); 2) the coreference/compatibility of argument is crucial for deciding coreference relations between events (Lee et al., 2012; Huang et al., 2019). Unfortunately, the traditional argument-based end-to-end pipeline event coreference methods (Chen and Ng, 2014; Yang et al., 2015) suffer from the error propagation problem of previous components, e.g., argument extraction and entity coreference. The denoising feature composition algorithms or joint modeling of entity/event coreference may effectively solve the argument's error propagation problem.

5 Related Work

Event Coreference. Event coreference aims to cluster textual mentions of the same event. Different from cross-document event coreference works (Yang et al., 2015; Zhang et al., 2015; Choubey and Huang, 2017; Kenyon-Dean et al., 2018; Barhom et al., 2019), this paper focuses on the within-document event coreference task.

Traditional approaches (Chen and Ji, 2009; Chen and Ng, 2014; Liu et al., 2014) are mostly pipeline-based systems depending on several upstream components, thus often suffer from the error propagation problem. To address this problem, many joint models have been proposed, e.g., joint inference (Chen and Ng, 2016; Lu et al., 2016) and joint modeling (Araki and Mitamura, 2015; Lu and Ng, 2017a). Furthermore, the above methods use hand-crafted features, which are hard to generalize to the new languages/domains/datasets. Several neural network models (Krause et al., 2016; Chao et al., 2019) and transfer techniques (Huang et al., 2019) are proposed to complement these methods with automatic feature learning abilities.

Compared to previous approaches, E^{3}C is the first fully end-to-end neural event coreference resolution approach. It can extract features, detect event mentions, and resolve event chains in the same network.

End-to-end Entity Coreference. Recently, end-to-end neural networks (Lee et al., 2017, 2018; Kantor and Globerson, 2019; Fei et al., 2019; Joshi et al., 2019) have achieved significant progress in entity coreference. These methods also motivate this study. Due to the mention diversity and the long-distance coreference problems, event coreference is usually considered more challenging than entity coreference (Lu and Ng, 2018; Choubey and Huang, 2018). This paper proposes a type-guided mechanism, where can resolve the above challenges by incorporating type information, learning semantic event mention representation, and modeling long-distance, semantic-dependent evidence.

6 Conclusions

This paper proposes a state-of-the-art, end-to-end neural network for event coreference resolution – E^{3}C neural network, which jointly models event detection and event coreference, and learns to extract features from the raw text directly. A type-guided mechanism is further proposed for resolving the mention diversity problem and the long-distance coreference problem, which: 1) informs coreference prediction with type scoring, 2) refines mention representation using type information, and 3) guides decoding under type consistency. Experiments show that our method achieves state-of-the-art performances on KBP 2016 and KBP 2017. For future work, we will focus on the bottleneck of event coreference, e.g., event detection and argument modeling.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Sydney, Australia. Association for Computational Linguistics.

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*, pages 2074–2080, Lisbon, Portugal. Association for Computational Linguistics.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Granada.

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*, pages 4179–4189, Florence, Italy. Association for Computational Linguistics.

Cosmin Bejan and Sanda Harabagiu. 2008. A linguistic resource for discovering event structures and resolving event coreference. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, Marrakech, Morocco.

Cosmin Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1412–1422, Uppsala, Sweden. Association for Computational Linguistics.

Wenhan Chao, Ping Wei, Zhunchen Luo, Xiao Liu, and Guobin Sui. 2019. Selective expression for event coreference resolution on twitter. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Bin Chen, Jian Su, Sinno Jialin Pan, and Chew Lim Tan. 2011. A unified event coreference resolution by integrating multiple resolvers. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 102–110, Chiang Mai, Thailand.

Chen Chen and Vincent Ng. 2014. SinoCoreferencer: An end-to-end Chinese event coreference resolver. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 4532–4538, Reykjavik, Iceland.

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 Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2913–2920. AAAI Press.

Zheng Chen and Heng Ji. 2009. Graph-based event coreference resolution. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 54–57, Suntec, Singapore. Association for Computational Linguistics.

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*, pages 2124–2133, Copenhagen, Denmark. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 485–495, Melbourne, Australia. Association for Computational Linguistics.

Agata Cybulska and Piek Vossen. 2013. Semantic relations between events and their time, locations and participants for event coreference resolution. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 156–163, Hissar, Bulgaria.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 4545–4552, Reykjavik, Iceland.

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

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *33rd Conference on Neural Information Processing Systems*, pages 13042–13054.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490.

Joe Ellis, Jeremy Getman, Dana Fore, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie Strassel. 2015. Overview of linguistic resources for the tac kbp 2015 evaluations: Methodologies and results. In *TAC 2015*.

Joe Ellis, Jeremy Getman, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie Strassel. 2016. Overview of linguistic resources for the tac kbp 2016 evaluations: Methodologies and results. In *TAC 2016*.

Hongliang Fei, Xu Li, Dingcheng Li, and Ping Li. 2019. End-to-end deep reinforcement learning based coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 660–665, Florence, Italy. Association for Computational Linguistics.

Jeremy Getman, Joe Ellis, Zhiyi Song, Jennifer Tracey, and Stephanie Strassel. 2017. Overview of linguistic resources for the tac kbp 2017 evaluations: Methodologies and results. In *TAC 2017*.

Kartik Goyal, Sujay Kumar Jauhar, Huiying Li, Mrinmaya Sachan, Shashank Srivastava, and Eduard Hovy. 2013. A structured distributional semantic model for event co-reference. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 467–473, Sofia, Bulgaria. Association for Computational Linguistics.

Eduard Hovy, Teruko Mitamura, Felisa Verdejo, Jun Araki, and Andrew Philpot. 2013. Events are not simple: Identity, non-identity, and quasi-identity. In *Workshop on Events: Definition, Detection, Coreference, and Representation*, pages 21–28, Atlanta, Georgia. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 785–795, Minneapolis, Minnesota. Association for Computational Linguistics.

Kevin Humphreys, Robert Gaizauskas, and Saliha Azzam. 1997. Event coreference for information extraction. In *Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*.

Shanshan Jiang, Yihan Li, Tianyi Qin, Qian Meng, and Bin Dong. 2017. Srcb entity discovery and linking (edl) and event nugget systems for tac 2017. In *TAC 2017*.

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5802–5807, Hong Kong, China. Association for Computational Linguistics.

Ben Kantor and Amir Globerson. 2019. Coreference resolution with entity equalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 673–677, Florence, Italy. Association for Computational Linguistics.

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 Seventh Joint Conference on Lexical and Computational Semantics*, pages 1–10, New Orleans, Louisiana. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *5th International Conference on Learning Representations*.

Sebastian Krause, Feiyu Xu, Hans Uszkoreit, and Dirk Weissenborn. 2016. Event linking with sentential features from convolutional neural networks. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 239–249, Berlin, Germany. Association for Computational Linguistics.

LDC. 2005. Ace (automatic content extraction) english annotation guidelines for events. Technical report, Linguistic Data Consortium.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Kenton Lee, Shimi Salant, Tom Kwiatkowski, Ankur Parikh, Dipanjan Das, and Jonathan Berant. 2016. Learning recurrent span representations for extractive question answering. *arXiv preprint arXiv:1611.01436*.

Zhengzhong Liu, Jun Araki, Eduard Hovy, and Teruko Mitamura. 2014. Supervised within-document event coreference using information propagation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 4539–4544, Reykjavik, Iceland.

Jing Lu and Vincent Ng. 2016a. Event coreference resolution with multi-pass sieves. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 3996–4003, Portorož, Slovenia.

Jing Lu and Vincent Ng. 2016b. Utds event nugget detection and coreference system at kbp 2016. In *TAC 2016*.

Jing Lu and Vincent Ng. 2017a. Joint learning for event coreference resolution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 90–101, Vancouver, Canada. Association for Computational Linguistics.

Jing Lu and Vincent Ng. 2017b. Learning antecedent structures for event coreference resolution. In *2017 16th IEEE International Conference on Machine Learning and Applications*, pages 113–118.

Jing Lu and Vincent Ng. 2018. Event coreference resolution: A survey of two decades of research. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 5479–5486.

Jing Lu, Deepak Venugopal, Vibhav Gogate, and Vincent Ng. 2016. Joint inference for event coreference resolution. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3264–3275, Osaka, Japan.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Katie McConky, Rakesh Nagi, Moises Sudit, and William Hughes. 2012. Improving event coreference by context extraction and dynamic feature weighting. In *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, pages 38–43. IEEE.

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*, pages 392–402, Austin, Texas. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Sameer S. Pradhan, Lance Ramshaw, Ralph Weischedel, Jessica MacBride, and Linnea Micciulla. 2007. Unrestricted coreference: Identifying entities and events in ontonotes. In *Proceedings of the International Conference on Semantic Computing*, pages 446–453, Washington, DC, USA.

Marta Recasens and Eduard Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural Language Engineering*, 17(4):485–510.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ERE: Annotation of entities, relations, and events. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98, Denver, Colorado. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *7th International Conference for Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *31st Conference on Neural Information Processing Systems*, pages 5998–6008.

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)*, pages 45–52, Columbia, Maryland.

Bishan Yang, Claire Cardie, and Peter Frazier. 2015. A hierarchical distance-dependent Bayesian model for event coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:517–528.

Tongtao Zhang, Hongzhi Li, Heng Ji, and Shih-Fu Chang. 2015. Cross-document event coreference resolution based on cross-media features. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 201–206, Lisbon, Portugal. Association for Computational Linguistics.

A Experiment Details

Table 6 presents the detailed hyper-parameters of the E^{3}C model used in our experiments. And we conducted all experiments on a Nvidia TITAN RTX GPU.

Parameter name	Parameter value
Mini batch size	1
Max epochs for stopping training	150
Patience for early stopping	10
Max antecedents number	50
Max document length for training	1024
Dropout for word representation	0.5
Dropout for FFNN	0.2
Hidden layers for FFNN	2
Hidden units for FFNN	150
Optimizer	Adamax
Initial learning rate	0.001
Learning rate anneal factor	0.5
Learning rate anneal patience	5

Table 6: Hyper-parameters of the E^{3}C model used in our experiments. FFNN indicates the feed-forward neural networks for mention proposaling and antecedent scoring.

B Data Sets

We used Stanford CoreNLP toolkit[9] to preprocess all documents for xml tags cleaning, sentence splitting and tokenization. Since only 18 categories were used for the official evaluation, we filtered out event instances with other categories in the training data.

C Reproducibility

In this section, we present the reproducibility information of the paper. Table 7 shows the corresponding validation performance for all reported KBP 2016 and KBP 2017 results. In addition, Table 8 presents the average runtime for each approach and number of parameters in each model.

[9] https://stanfordnlp.github.io/CoreNLP/

		Type-P	**Type-R**	**Type-F1**	**B_3**	**CEAF$_e$**	**MUC**	**BLANC**	**AVG-F**
KBP 2016	Interact Model$_{\text{BERT}}$	56.8	59.18	57.97	45.49	44.07	37.43	30.41	39.35
	E^{3}C	62.94	59.10	60.96	49.02	46.76	42.80	33.00	42.89
KBP 2017	Interact Model$_{\text{BERT}}$	60.97	57.36	59.11	49.72	50.30	32.50	32.06	41.14
	E^{3}C	65.85	54.61	59.71	51.60	51.48	39.45	35.01	44.38
	E^{3}C$_{\text{w/o Type-Refined}}$	70.33	48.64	57.51	50.45	49.96	40.15	34.16	43.68
	E^{3}C$_{\text{w/o Type-Guided}}$	65.97	54.93	59.94	47.26	41.93	37.22	33.54	39.99
	E^{3}C$_{\text{Two Stage}}$	63.92	55.26	59.27	51.08	49.80	35.71	33.91	42.62
	E^{3}C$_{\text{w/o Proposal Loss}}$	68.15	51.33	58.56	50.80	51.29	37.05	34.14	43.32
	E^{3}C$_{\text{GloVe+Char}}$	63.92	50.50	56.43	47.86	45.76	36.45	31.92	40.50
	E^{3}C$_{\text{ELMo}}$	64.42	53.74	58.60	50.57	50.70	35.25	33.42	42.48
	E^{3}C$_{\text{BERT-BASE-Cased}}$	68.08	52.75	59.45	51.46	50.53	38.66	34.85	43.88
	E^{3}C$_{\text{BERT-LARGE-Uncased}}$	66.65	55.97	60.85	53.21	52.60	41.69	37.37	46.22
	E^{3}C$_{\text{BERT-LARGE-WWM-Uncased}}$	69.48	55.01	61.40	53.78	53.01	41.62	36.80	46.30
	Type-Rule	65.85	54.61	59.71	35.88	26.91	29.39	25.91	29.52
	E^{3}C$_{\text{GoldMention}}$	81.39	70.95	75.81	67.76	66.41	48.16	51.60	58.48

Table 7: Corresponding validation performance for each reported KBP 2016/2017 result. Type-Rule and E^{3}C$_{\text{GoldMention}}$ take unreal experiment setups for exploiting the bound performance of E^{3}C. Type Rule is a simple heuristic method that regards all event mentions in the same type are coreferential, and it directly uses the event detection result from E^{3}C. E^{3}C$_{\text{GoldMention}}$ uses gold mentions instead of mentions proposed by the mention proposal layer, but the model still needs to predict the type of each mention.

	Time for one epoch (s)	$\|\mathbf{\Theta}_{\text{update}}\|$
E^{3}C	82.76	2,886,108
Interact Model$_{\text{BERT}}$	408.25	3,613,431
E^{3}C$_{\text{w/o Type-Refined}}$	78.77	1,705,692
E^{3}C$_{\text{w/o Type-Guided}}$	81.80	2,886,108
E^{3}C$_{\text{Two Stage}}$	75.52	1,708,561
E^{3}C$_{\text{w/o Proposal Loss}}$	80.33	2,886,108
E^{3}C$_{\text{GloVe + Char}}$	100.00	2,272,595
E^{3}C$_{\text{ELMo}}$	1345.30	4,880,339
E^{3}C$_{\text{BERT-BASE-Cased}}$	82.74	2,886,108
E^{3}C$_{\text{BERT-LARGE-Uncased}}$	152.23	4,880,360
E^{3}C$_{\text{BERT-LARGE-WWM-Uncased}}$	152.47	4,880,360

Table 8: Average runtime for each approach and number of parameters in each model. $\mathbf{\Theta}_{\text{update}}$ refers to the number of trainable parameters. We fix all parameters for word representations in our experiments, such as BERT, GloVe, and ELMo parameters.