Event Coreference Resolution with Non-Local Information

Jing Lu and **Vincent Ng**
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
{ljwinnie,vince}@hlt.utdallas.edu

Abstract

Existing event coreference resolvers have largely focused on exploiting the information extracted from the local contexts of the event mentions under consideration. Hypothesizing that non-local information could also be useful for event coreference resolution, we present two extensions to a state-of-the-art joint event coreference model that involve incorporating (1) a supervised topic model for improving trigger detection by providing global context, and (2) a preprocessing module that seeks to improve event coreference by discarding unlikely candidate antecedents of an event mention using discourse contexts computed based on salient entities. The resulting model yields the best results reported to date on the KBP 2017 English and Chinese datasets.

1 Introduction

Event coreference resolution is the task of determining the event mentions in a document that refer to the same real-world event. One of its major challenges concerns error propagation: since the event coreference resolution component typically lies towards the end of the standard information extraction pipeline, the performance of an event coreference resolver can be adversely affected by errors propagated from its upstream components. The upstream component that has the largest impact on event coreference performance is arguably *trigger detection*. Recall that the goal of a trigger detector is to identify event triggers and assign an event subtype to each of them. Failure to detect triggers could therefore limit the upper bound on event coreference performance.

To address error propagation, one way that has been shown to be effective for a variety of NLP tasks is to develop *joint* models, which allow cross-task output constraints to be learned from annotated training data. For event coreference, a learner can easily learn, for instance, that two coreferent event mentions must have the same event subtype, thereby allowing event coreference to influence trigger detection. Unfortunately, the vast majority of existing event coreference resolvers have adopted a pipeline architecture where trigger detection precedes event coreference. In particular, joint models are both under-studied and under-exploited for event coreference given the usefulness they have demonstrated for other NLP tasks. One exception is Lu and Ng's (2017a) joint model, which jointly learns trigger detection and event coreference and has achieved state-of-the-art results. As a structured conditional random field, the model employs unary factors to encode the features specific for each task and binary/ternary factors to capture the interaction between each pair of tasks. The use of binary/ternary factors is a particularly appealing aspect of this model: it allows these cross-task interactions to be captured in a *soft* manner, enabling the learner to *learn* which combinations of values of the output variables are more probable.

We hypothesize that the power of this joint event coreference model has not been fully exploited and seek to extend it in this paper. Our extensions are based on the observation that the strength of a joint model stems from its ability to facilitate cross-task knowledge transfer. In other words, the better we can model *each* task involved, the more we can potentially get out of joint modeling. Given this observation, we seek to improve the modeling of these tasks in this joint model as follows.

First, we improve trigger detection by exploiting topic information. State-of-the-art trigger detectors, including those based on deep neural networks (e.g., Nguyen et al. (2016)), classify each candidate trigger using local information and largely ignore the fact that the *topic* of the document in which a trigger appears plays an important role in determining its event subtype. To understand the usefulness

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 653–663
December 4 - 7, 2020. ©2020 Association for Computational Linguistics

Three journalists at The New York Times on Tuesday announced plans to {leave}$_{ev1}$ the newspaper. The {departures}$_{ev2}$ follow moves last month by several other Times employees, all of whom were {leaving}$_{ev3}$ to join digital companies.
Pakistan's Interior Ministry has ordered New York Times Reporter to {leave}$_{ev4}$. The ministry gave no explanation for the expulsion order. "You are therefore advised to {leave}$_{ev5}$ the country within 72 hours," the order stated.

Table 1: Event coreference resolution examples.

of document topics, consider the examples in Table 1: although all five events have similar trigger words, we can see that the meaning of the triggers and their event subtypes are different in different contexts. Hence, if an event coreference model knows that the topics of these two documents are different, it can exploit this information to more accurately classify their event subtypes. In particular, we propose to train a supervised topic model to infer the topic of each word in a test document, with the goal of understanding each candidate trigger using its *global* in addition to local context.

Second, we improve event coreference by exploiting *discourse* information. Specifically, we introduce a *preprocessing* component for event coreference resolution where we *prune* the candidate antecedents of an event mention that are unlikely to be its correct antecedent based on *discourse context*. In essence, this discourse-based preprocessing step seeks to simplify the job of the event coreference model by reducing the number of candidate antecedents it has to consider for a given event mention. We encode the discourse context of an event mention using the entities that are *salient* at the point of the discourse in which the event mention appears. To our knowledge, we are the first to show that event coreference performance can be improved using discourse contexts that are encoded using salient discourse entities.

In sum, the contributions of this paper are twofold. First, while existing event coreference resolvers have largely focused on exploiting the information extracted from the local contexts of the event mentions under consideration, we show how a state-of-the-art joint event coreference model can be improved using the *non-local* information provided by a supervised topic model and salient discourse entities. Second, the resulting model achieves the best results to date on the KBP 2017 English and Chinese event coreference datasets.

2 Definitions and Corpora

2.1 Definitions

We employ the following definitions in our discussion of trigger detection and event coreference:

- An **event trigger** is a string of text that most clearly expresses the occurrence of an event, usually a word or a multi-word phrase.
- An **event mention** is an explicit occurrence of an event consisting of a textual trigger, arguments or participants (if any), and the event type/subtype.
- An **event coreference chain** (a.k.a. an **event hopper**) is a group of event mentions that refer to the same real-world event. They must have the same event (sub)type.

To understand these definitions, consider the example in Table 1, which contains five event mentions from two documents. The first one consists of three event mentions of subtype Personnel.Endposition, among which $ev1$ and $ev2$, which are triggered by "leave" and "departures" respectively, are coreferent since they describe the event that three journalists resign. The second one consists of two coreferent event mentions, $ev4$ and $ev5$, both of which are triggered by "leave" and have subtype Movement.Transport_Person.

2.2 Corpora

We employ the English and Chinese corpora used in the TAC KBP 2017 Event Nugget Detection and Coreference task for evaluation, which are composed of two types of documents, newswire documents and discussion forum documents. There are no official training sets: the task organizers have simply made available a number of event coreference-annotated corpora for training. For English, we use LDC2015E29, E68, E73, E94, and LDC2016E64 for training. Together they contain 817 documents with 22894 event mentions distributed over 13146 coreference chains. For Chinese, we use LDC2015E78, E105, E112, and LDC2016E64 for training. Together they contain 548 documents with 7388 event mentions distributed over 5526 coreference chains.

The KBP 2017 English test set consists of 167 documents with 4375 event mentions distributed over 2963 coreference chains. The Chinese test set consists of 167 documents with 3884 event mentions distributed over 2558 coreference chains.

3 Model

Following Lu and Ng (2017a), we employ a structured conditional random field, which operates at the document level. Specifically, given a test document, we first extract from it all single- and multi-word nouns and verbs that have appeared at least once as a trigger in the training data. We treat each of these extracted nouns and verbs as a candidate event mention. The goal of the model is to make joint predictions for the candidate event mentions in a document. Three predictions will be made for each candidate event mention that correspond to the three tasks in the model: its trigger subtype, its induced topic, and its antecedent.

Given this formulation, we define three types of output variables. The first type consists of event subtype variables $\mathbf{s} = (s_1, \ldots, s_n)$. Each s_i takes a value in the set of the 18 event subtypes defined in KBP 2017 or NONE, which indicates that the event mention is not a trigger. The second type consists of coreference variables $\mathbf{c} = (c_1, \ldots, c_n)$, where $c_i \in \{1, \ldots, i-1, \text{NEW}\}$. In other words, the value of each c_i is the id of its antecedent, which can be one of the preceding event mentions, or NEW (if the mention underlying c_i starts a new cluster). The third type consists of topic variables $\mathbf{t} = (t_1, \ldots, t_n)$. Each t_i takes a value in a 19-element set in which the topics have a one-to-one correspondence with the event subtype labels defined above. Despite this one-to-one mapping, these two types of labels should not be interpreted in the same manner. As we will see, a word's induced topic label is influenced by our supervised topic model, whereas a word's subtype is not.

Each candidate event mention is associated with one coreference variable, one event subtype variable, and one topic variable. Our model induces a probability distribution over these variables:

$$p(\mathbf{s}, \mathbf{c}, \mathbf{t} | x; \Theta) \propto \exp(\sum_i \theta_i f_i(\mathbf{s}, \mathbf{c}, \mathbf{t}, x))$$

where $\theta_i \in \Theta$ is the weight associated with feature function f_i and x is the input document.

3.1 Independent Models

3.1.1 Trigger Detection Model

Each instance for training the trigger detection model corresponds to a candidate trigger in the training set, which is created as follows. For each word w that appears as a true trigger at least once in the training data, we create a candidate trigger from each occurrence of w in the training data. If a given occurrence of w is a true trigger in the associated document, the class label of the corresponding training instance is its subtype label. Otherwise, we label the instance as NONE.

Each candidate trigger m is represented using features generated from the following feature templates: m's word, m's lemma, word bigrams formed with a window size of three from m; feature conjunctions created by pairing m's lemma with each of the following features: the head word of the entity syntactically closest to m, the head word of the entity textually closest to m, the entity type of the entity that is syntactically closest to m, and the entity type of the entity that is textually closest to m.[1] In addition, for event mentions with verb triggers, we use the head words and the entity types of their subjects and objects as features, where the subjects and objects are extracted from the dependency parses produced by Stanford CoreNLP (Manning et al., 2014). For event mentions with noun triggers, we create the same features except that we replace the subjects and verbs with heuristically extracted agents and patients.

3.1.2 Topic Model

Our first extension to Lu and Ng's (2017a) model seeks to improve trigger detection using topic information. We train a supervised topic model to infer the topic of each word in a test document, with the goal of understanding each candidate trigger using its *global* in addition to local context.

Like the trigger detection model, each training instance corresponds to a candidate trigger. The class label is the topic label of the candidate trigger. We have 19 topic labels in total: there is a one-to-one correspondence between the 18 subtype labels and 18 of the topic labels. The remaining topic label is OTHER, which is reserved for those words that do not belong to any of the 18 topics. Topic labels can be derived directly from subtype labels given the one-to-one correspondence between them. Each candidate trigger is represented using 19 features, which correspond to the 19 topic labels. The value of a feature, which is derived from the output of a LabeledLDA model (Ramage et al., 2009), encodes the probability that the candidate trigger belongs to the corresponding topic.

To train the LabeledLDA model, we first apply LabeledLDA using the Mallet toolkit (McCallum,

[1] We use an in-house CRF-based entity extraction model to jointly identify the entity mentions and their types.

2002) to the training documents, which learns a distribution over words for each topic, β. We represent each training document using the candidate triggers as well as the context words that are useful for distinguishing the topics.[2] To get the useful context words, we rank the words in the training documents by their weighted log-likelihood ratios:

$$P(w_i|m_j, v_k) \log \frac{P(w_i|m_j, v_k)}{P(w_i|m_j, \neg v_k)}$$

where w_i, m_j and v_k denote the ith word in the vocabulary, the jth candidate trigger word and the kth subtype (including NONE), respectively. Intuitively, a word w_i will have a high rank with respect to a candidate trigger word m_j of subtype v_k if it appears frequently with m_j of subtype v_k and infrequently with m_j of other subtypes. We employ as the useful context words the top 125 words ranked by the weighted log likelihood ratio w.r.t. each pair of trigger and subtype. The label set of each training document is the set of subtypes collected from all the triggers in the document plus NONE.

After training, we apply the resulting LabeledLDA model to a test document, which is represented using the candidate triggers and the useful context words, as defined above. Specifically, given a test document, we (1) apply the model to infer the distribution of topics in the document, and then (2) compute the posterior distribution of topics given each candidate trigger in the document using Bayes rule as follows:

$$P(z|m) \propto P(m|z : \beta)P(z)$$

where $P(z)$ is the distribution of topic z in the test document, $P(m|z : \beta)$ is the topic-dependent distribution of candidate triggers m that is learned from the training documents, and $P(z|m)$ is the posterior distribution of z given m in the test document. We use this posterior distribution to generate features for representing each instance for training/testing the topic model, as described above.

Note that while the label sets used by the trigger detector and the topic model are functionally equivalent, they are trained using different feature sets. The features used by the trigger detector encodes a candidate trigger's local context, while the features used by the topic model encodes its global context (e.g., its relationship with other words).

3.1.3 Event Coreference Model

Our event coreference model is an adaptation of Durrett and Klein's (2013) mention-ranking model, which was originally developed for entity coreference, to the task of event coreference. This model selects the most probable antecedent for a mention to be resolved from its set of candidate antecedents (or NEW if the mention is non-anaphoric).

We employ two types of feature templates to represent the candidate antecedents for the event mention to be resolved, m_j. The first type is composed of features that represent the NULL candidate antecedent.[3] These include: m_j's word, m_j's lemma, a conjoined feature created by pairing m_j's lemma with the number of sentences preceding m_j, and another conjoined feature created by pairing m_j's lemma with the number of mentions preceding m_j in the document. The second type is composed of features that represent a non-NULL candidate antecedent, m_i. These include m_i's word, m_i's lemma, whether m_i and m_j have the same lemma, and the following feature conjunctions: (1) m_i's word paired with m_j's word, (2) m_i's lemma paired with m_j's lemma, (3) the sentence distance between m_i and m_j paired with m_i's lemma and m_j's lemma, (4) the mention distance between m_i and m_j paired with m_i's lemma and m_j's lemma, (5) a quadruple consisting of m_i and m_j's subjects and their lemmas, and (6) a quadruple consisting of m_i and m_j's objects and their lemmas.

Our second extension to Lu and Ng's (2017a) model involves leveraging *discourse* information to improve this event coreference model. Specifically, we introduce a *preprocessing* component for event coreference resolution where we *prune* the candidate antecedents of an event mention that are unlikely to be its correct antecedent based on *discourse context*. The idea is to (1) encode the discourse context of each event mention in a document using the *entities* that are *salient* at the point of the discourse in which the event mention appears, and by hypothesizing that two event mentions that appear in different discourse contexts are unlikely to be coreferent, we (2) prune any candidate antecedent of an event mention m whose discourse context is different from that of m, allowing the event coreference model to resolve an event mention to one of the candidate antecedents that survive this discourse-based filtering step. In essence, this

[2] If a candidate trigger is a multi-word phrase, we treat it as a "word" by concatenating its constituent words using underscores (e.g.,"step down" is represented as "step_down").

[3] Resolving a mention to the NULL antecedent is the same as having the mention starts a NEW cluster.

preprocessing step seeks to simplify the job of the event coreference model by reducing the number of candidate antecedents it has to consider for a given event mention.

Since we aim to encode the discourse context of each event mention using the entities that are salient at the point of the discourse in which the event mention appears, we need to compute the salience score of each entity E w.r.t. each event mention m. We employ the following formula, which was proposed by Chen and Ng (2015b):

$$\sum_{e \in E} g(e) \times decay(e)$$

In this formula, e is a mention of entity E that appears in either the same sentence as m or one of its preceding sentences. $g(e)$ is a score that is computed based on the grammatical role of e in the sentence: 4 if e is a subject, 2 if it is an object, and 1 otherwise. $decay(e)$ is a decay factor that is set to 0.5^{dis}, where dis is the sentence distance between e and m. We compute discourse entities using Stanford CoreNLP's neural entity coreference resolver and grammatical roles using CoreNLP's syntactic dependency parser.

Next, we define the discourse context of an event mention m to be the list of entities whose salience score is at least 1 when computed w.r.t. m. As noted before, we aim to prune the unlikely candidate antecedents of an event mention m, namely those candidates whose discourse contexts are different from that of m. Rather than heuristically defining a function for computing the similarity between two different discourse contexts, we train a ranker that ranks the candidate antecedents of m based on two types of features derived from their discourse contexts:

Salience score ratios (SSRs): For each entity E that appears in the discourse contexts of both candidate antecedent c and m, we first compute E's SSR as the ratio of E's salience score computed w.r.t. m to E's salience score computed w.r.t. c. (If this ratio is less than 1, we take its reciprocal.) Then, for each (c, m) pair, we create five features that encode the number of entities whose SSR falls into each of these five intervals: [1,1], (1, 2], (2, 3], (3,4], (4,5], and [5, inf]. Intuitively, c's and m's discourse contexts tend to be more similar if they have more entities in the lower buckets.

Lexical features: For each mention em_1 of each entity in candidate antecedent c's discourse context and each mention em_2 of each entity in m's discourse context, we create a lexical feature that pairs em_1's head with em_2's head.

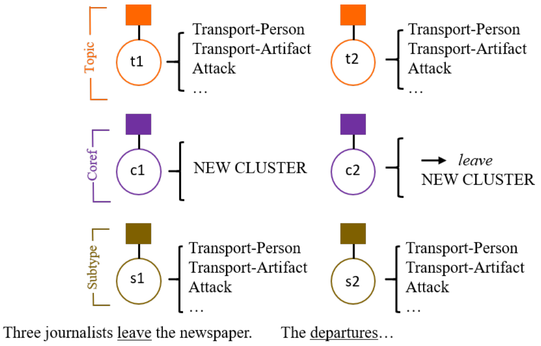

Figure 1: Unary factors for the three tasks, the variables they are connected to, and the possible values of the variables.

To train this ranker, we employ the same log-linear model as the one used for the event coreference model, where the training objective is to maximize the likelihood of selecting the correct antecedent for each event mention.

After training, we apply this ranker to prune all but the top k candidate antecedents of each event mention in a test document. These k candidate antecedents, together with the NULL candidate antecedent, will be ranked by the event coreference model, and the highest-ranked candidate will be selected as the antecedent of the event mention under consideration.[4] We treat k as a hyperparameter and tune it on the development set.

It is worth noting that we prune the candidate antecedents of the event mentions not only in the test set but also in the training set. We produce the top k candidate antecedents of each event mention in the training set via five-fold cross-validation over the training documents.

Figure 1 illustrates the unary factors, which encode the features used in the three independent models. Specifically, the sentence fragment at the bottom of the figure contains two event mentions, one triggered by *leave* and the other by *departure*. Each of them is associated with three variables, one for each of the three models. Next to each variable is the set of possible values of that variable.

3.2 Joint Learning

To perform joint training over the three models described in the previous subsection, we need to

[4] The discourse preprocessing module does not handle NULL candidate antecedents, so they will always be available to the event coreference model.

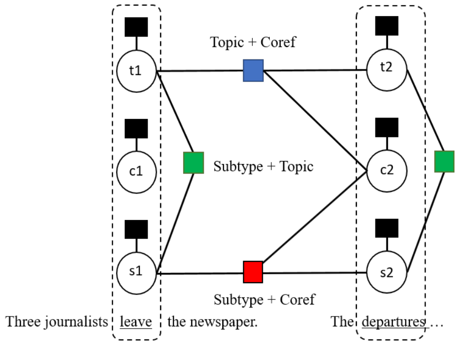

Figure 2: Binary and ternary factors.

define (1) features that capture the interaction between the two tasks, (2) the joint training scheme, and (3) the inference mechanism.

3.2.1 Cross-Task Interaction Features

Our cross-task interaction features, which capture the *pairwise* interaction between our tasks, are associated with ternary factors, as described below.

Trigger detection and coreference. We define our joint coreference and trigger detection factors such that the features defined on subtype variables s_i and s_j are fired only if current mention m_j is coreferent with preceding mention m_i. These features are: (1) the pair of m_i and m_j's subtypes; (2) the pair of m_j's subtype and m_i's word; and (3) the pair of m_i's subtype and m_j's word.

Trigger detection and topic modeling. We fire features (encoded as binary factors) that conjoin each candidate event mention's event subtype, its topic and the lemma of its trigger.

Topic modeling and coreference. Our joint coreference and topic modeling factors and features are the same as those for trigger detection and coreference, except that event subtype labels are replaced with topic labels. In other words, the features are defined on the topic labels.

Figure 2 shows the cross-task interaction features. The green factor is binary, connecting a subtype variable and a topic variable. The red factor is ternary, connecting two subtype variables to a coreference variable. Finally, the blue factor is also ternary, connecting topic with coreference.

3.2.2 Training

The joint training scheme seeks to learn the model parameters Θ from a set of d training documents, where document i contains content x_i, gold trigger annotations $\mathbf{s_i^*}$, topic labels $\mathbf{t_i^*}$ inferred from the LabeledLDA model using Gibbs sampling, and gold event coreference partition C_i^*, by maximizing the following conditional likelihood of the training data with L_1 regularization:[5]

$$L(\Theta) = \sum_{i=1}^{d} \log \sum_{\mathbf{c}^* \in A(C_i^*)} p'(\mathbf{s_i^*}, \mathbf{t_i^*}, \mathbf{c}^* | x_i; \Theta) + \lambda \|\Theta\|_1$$

where $p'(\mathbf{s}^*, \mathbf{t}^*, \mathbf{c}^* | x; \Theta)$ is $p(\mathbf{s}^*, \mathbf{t}^*, \mathbf{c}^* | x; \Theta)$ augmented with task-specific loss functions. Specifically,

$$p'(\mathbf{s}^*, \mathbf{t}^*, \mathbf{c}^* | x; \Theta) \propto p(\mathbf{s}^*, \mathbf{t}^*, \mathbf{c}^* | x; \Theta) \exp[\\ \alpha_s l_s(\mathbf{s}, \mathbf{s}^*) + \alpha_t l_t(\mathbf{t}, \mathbf{t}^*) + \alpha_c l_c(\mathbf{c}, C^*)]$$

where l_s, l_t and l_c are task-specific loss functions[6], and α_s, α_t and α_c are the associated weight parameters that specify the relative importance of the three tasks in the objective function.[7] We use AdaGrad (Duchi et al., 2011) to optimize our objective function with $\lambda = 0.001$.

3.2.3 Inference

Inference, which is performed during training and decoding, involves computing the marginals for a variable or a set of variables to which a factor connects. For efficiency, we perform approximate inference using belief propagation, running it until convergence. We use minimum Bayes risk decoding, where we compute the marginals for each variable in our model and independently return the most likely setting of each variable. Marginals typically converge in 3–5 iterations of belief propagation, so we use 5 iterations in our experiments.

4 Evaluation

4.1 Experimental Setup

We perform training and evaluation on the KBP 2017 English and Chinese corpora. For English,

[5] In the conditional log likelihood function, $A(C_i^*)$ is the set of antecedent structures that are consistent with C_i^*. Since our model needs to be trained on antecedent vectors $\mathbf{c}^*$ but the gold coreference annotation for each document i is provided in the form of a clustering C_i^*, we need to sum over all consistent antecedent structures.

[6] The loss function for event coreference, which is introduced by Durrett and Klein (2013) for entity coreference resolution, is a weighted sum of (1) the number of anaphoric mentions misclassified as non-anaphoric, (2) the number of non-anaphoric mentions misclassified as anaphoric, and (3) the number of incorrectly resolved mentions. The loss function for trigger detection is parameterized in a similar way, having three parameters associated with (1) the number of non-triggers misclassified as triggers, (2) the number of triggers misclassified as non-triggers, and (3) the number of triggers labeled with the wrong subtype. The loss function for topic detection is defined in a similar way as trigger detection.

[7] These weight parameters, as well as those that are used within the loss functions, are tuned on the development set using grid search.

		Event Coreference						Trigger Detection			
	English	MUC	B^3	CEAF$_e$	BLANC	AVG-F	Δ	P	R	F	Δ
1	Huang et al. (2019)	35.7	43.2	40.0	32.4	36.8		56.8	46.4	51.1	
2	Full	37.11	44.49	40.03	29.93	**37.89**		64.45	46.92	**54.30**	
3	− Topic	34.16	43.76	40.78	28.20	36.72	−1.17	64.39	46.67	54.11	−0.19
4	− Discourse	34.53	43.06	40.07	27.95	36.40	−1.49	62.15	47.49	53.84	−0.46
5	− Both	31.94	42.84	40.21	26.49	35.37	−2.52	63.57	45.87	53.29	−0.89
		Event Coreference						Trigger Detection			
	Chinese	MUC	B^3	CEAF$_e$	BLANC	AVG-F	Δ	P	R	F	Δ
6	Lu and Ng (2017b)	27.07	34.18	32.22	18.57	28.01		46.61	46.91	46.76	
7	Full	27.89	40.95	39.49	22.00	**32.58**		51.81	54.81	**53.27**	
8	− Topic	26.39	40.43	38.75	21.18	31.69	−0.89	51.81	53.28	52.53	−0.74
9	− Discourse	26.13	40.78	39.31	21.02	31.81	−0.77	51.65	54.65	53.11	−0.16
10	− Both	25.93	37.50	34.24	19.92	29.40	−3.18	56.78	44.63	49.98	−3.29

Table 2: Results of event coreference and trigger detection on the KBP 2017 English and Chinese test sets. Baseline results (rows 1 and 6) are copied verbatim from the original papers.

we train models on 646 of the training documents, tune parameters on 171 training documents, and report results on the official KBP 2017 English test set. For Chinese, we train models on 438 of the training documents, tune parameters on 110 training documents, and report results on the official KBP 2017 Chinese test set.

Results of event coreference and trigger detection are obtained using version 1.8 of the official scorer provided by the KBP 2017 organizers. To evaluate event coreference performance, the scorer employs four commonly-used scoring measures, namely MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), CEAF$_e$ (Luo, 2005) and BLANC (Recasens and Hovy, 2011), as well as the unweighted average of their F-scores (AVG-F). The scorer reports event mention detection performance in terms of Precision (P), Recall (R) and F-score, considering a mention correctly detected if it has an exact match with a gold mention in terms of boundary and event subtype.

4.2 Results

Results on the English test set are shown in the top half of Table 2. Specifically, row 1 shows the results of Huang et al.'s (2019) resolver, which has produced best results to date on this test set. Row 2 shows the results of our full model, which substantially outperforms the baseline system (row 1), yielding an improvement of 1.09 points in AVG-F for event coreference and 3.2 points in F-score for trigger detection. Note that the improvement in the MUC and B^3 F-scores is largely offset by the precipitation in the BLANC F-score.

Results on the Chinese test set are shown in the bottom half of Table 2. Specifically, row 6 shows the results of Lu and Ng's (2017b) resolver, which is the top KBP 2017 system for Chinese and has produced the best results to date on this test set. Our full model (row 7) outperforms this baseline by 4.57 points in AVG-F for event coreference and 6.51 points in F-score for trigger detection. Despite the large improvement in AVG-F, the MUC F-score only increases by 0.82 points. Since MUC F-scores are computed solely based on coreference links, these results suggest that the improvement in AVG-F can largely be attributed to successful identification singleton clusters rather than successful identification of coreference links.

4.3 Model Ablations

To evaluate the importance of each of the two extensions in the full model, we perform ablation experiments. Rows 3–5 and rows 8–10 in Table 2 show the English and Chinese results obtained using models that are retrained after one or both of the extensions are removed from the full model. The changes in AVG-F as a result of the ablations are shown in the Δ columns for both tasks.

Similar conclusions can be drawn from the ablation results for both languages. First, ablating each of the two extensions causes a drop in performance for both event coreference and trigger detection. These results suggest that topic modeling and discourse pruning are both useful for the two tasks. Second, ablating both extensions causes a more abrupt drop in performance than ablating one of the extensions. This implies that each extension is providing useful information for each task that cannot be provided by the other extension. Third, when both extensions are ablated, the resulting models still outperform the baselines for both tasks. Nevertheless, we can see that for English, discourse pruning contributes more to the performance of our full model than topic modeling, whereas the reverse is true for Chinese.

		English		Chinese	
		Training	**Test**	**Training**	**Test**
1	Number of candidate event mentions to be resolved	52370	9494	39758	9918
2	Number of candidate antecedents **before** pruning	371718	48750	124292	26406
3	Number of candidate antecedents **after** pruning	119416	20956	83378	20109
4	Number (%) of anaphoric event mentions	4362 (8.3%)	914 (9.6%)	1713 (4.3%)	821 (8.3%)
5	Number (%) of anaphoric event mentions whose correct antecedent are among the candidates **before** pruning	4317 (99.0%)	803 (87.8%)	1671 (97.6%)	585 (71.3%)
6	Number (%) of anaphoric event mentions whose correct antecedent are among the candidates **after** pruning	3171 (72.7%)	670 (73.3%)	1610 (94.0%)	565 (68.8%)

Table 3: Statistics on salience-based candidate pruning.

4.4 Analysis of Salience-Based Pruning

To gain insights into the effectiveness of discourse modeling in terms of pruning candidate antecedents, Table 3 shows some statistics on the candidate antecedents before and after applying pruning. Concretely, row 1 shows the total number of event mentions to be resolved in the English and Chinese training and test sets. For English, as we can see in rows 2–3, only 32.1% and 43.0% of the candidate antecedents remain in the training and test sets respectively after pruning. This can be attributed to the fact that we aggressively prune the candidate antecedents by allowing k (the number of top candidate antecedents that can survive the pruning for each event mention) to be in the range of 1 to 5 during parameter tuning.[8] Row 4 shows that among all event mentions to be resolved, only 8.3% of them are anaphoric. Row 5 shows that before pruning, the correct antecedent of almost all of the anaphoric event mentions in the training set is among the set of candidate antecedents, whereas the corresponding number on the test set is only 87.8% due to the presence of unseen event mentions. Row 6 shows that 72.7% and 73.3% of the correct antecedents on the training set and the test set survive the pruning, respectively. Similar trends can be observed for the Chinese datasets. Overall, these statistics shed light on why discourse-based pruning is beneficial: the percentage of correct antecedents that survive the pruning is far greater than the percentage of candidate antecedents that are pruned.

4.5 Discussion

One thing that the reader may not be able to appreciate just by looking at the performance numbers in Table 2 is that our two extensions are starting to attack some of the non-trivial aspects of event coreference that involve semantics and discourse, as opposed to those previous approaches that focus on low-level issues (e.g., string matching). For this reason, we will take a look at some of the errors addressed by our extensions below.

Let us first consider the kind of errors topic modeling allows us to address. Consider the first two sentences in Table 4, both of which contain the trigger candidate "struck". While "struck" triggers a "Conflict.Attack" event in the first sentence, neither of its occurrences in the second sentence corresponds to a true trigger (and therefore their subtypes should both be NONE). Without topic modeling, the model predicts all occurrences of "struck" in these sentences as belonging to Conflict.Attack (and hence misclassifies the subtypes of m_2 and m_3). The reasons are that (1) "struck" is most frequently associated with "Conflict.Attack" in the training data, and (2) since the two sentences have a similar syntactic structure and contain entities of the same type, the model fails to identify their differences. In contrast, with topic modeling, our model correctly predicts the topic of the document in which the second example appears as Contact.Meeting. Since the model manages to learn that the subtype of "struck" should be NONE when the topic is Contact.Meeting and that its subtype should be "Conflict.Attack" when the topic is "Conflict.Attack", it correctly predicts m_2 and m_3 as having subtype NONE and, as a result, it also correctly determines that they are not coreferent. In other words, by using global information encoded by the topic model, our model can distinguish between words that have different meanings in different contexts.

Next, consider the last example in Table 4, which aims to give the reader an idea of the usefulness of discourse-based pruning. In this example, m_4, m_5, and m_8 refer to the event of the French soldier being stabbed and are coreferent, whereas m_6 and m_7

[8] The best k according to the development set is 2 for English and 3 for Chinese.

A barrage of US missile {**struck**}$_{m_1}$ Pakistan's North Waziristan tribal district on Tuesday, killing at least 15 militants.
President Vladimir Putin sent his condolences to U.S. President Barack Obama on Tuesday over the deadly tornado that {**struck**}$_{m_2}$ Oaklahoma City. The tornado {**struck**}$_{m_3}$ the southern suburbs of the Oklahoma state capital Monday afternoon, killing at least 51 people and injuring at least 140 others.
The French police said they were continuing to search for the man responsible for {**stabbing**}$_{m_4}$ a uniformed soldier in the neck Saturday evening. The soldier was {**stabbed**}$_{m_5}$ in the back of the neck with a box cutter or short knife as he patrolled with two colleagues through the transport station of La Défense, a business area in a suburb of Paris. The police suggested that the deed may have been inspired by the {**attack**}$_{m_6}$ on a British soldier in a London street Wednesday. A spokesman for the police union UNSA, Christophe Crépin, said there were similarities with the London {**attack**}$_{m_7}$. The case of the {**wounded**}$_{m_8}$ soldier, Pfc. Cédric Cordier, 23, is being handled by France's anti-terrorism court, officials said Sunday.

Table 4: Examples illustrating the usefulness of topic modeling and salience-based pruning.

refer to the attack on the British solider and form another coreference cluster. Without discourse-based pruning, the model mistakenly links m_8 with m_7 because they both have subtype "Conflict.Attack". In contrast, discourse-based pruning ranks m_4 and m_5 higher than m_6 and m_7 in m_8's list of candidate antecedents, the reason being that m_4, m_5, and m_8 share the same entity (realized as "a uniformed soldier", "The soldier", and "the wounded soldier") in their contexts. Since the model retains only the top two candidate antecedents for English, m_6 and m_7 are being pruned, and the model successfully resolves m_8 to m_5.

5 Related Work

Using topics and salience. For event coreference, the notion of "topics" has thus far been exploited only for *cross-document* event coreference, where documents are clustered by topics so that no cross-document coreference links can be established between documents in different clusters (Lee et al., 2012; Choubey and Huang, 2017). These resolvers, unlike ours, are pipelined systems, meaning that topic detection can influence event coreference resolution but not the other way round. As for discourse salience, we are not aware of any event coreference work that attempts to explicitly model it, although one can argue that existing systems may have implicitly encoded it in a shallow manner via exploiting features that encode the distance between two event mentions (Liu et al., 2014; Cybulska and Vossen, 2015).

Computing argument compatibility. In addition to discourse-based pruning, candidate antecedents can be pruned based on how compatible the arguments of the two event mentions are. To capture argument compatibility, argument features have been extensively exploited. Basic features such as the number of overlapping arguments and the number of unique arguments, and a binary feature encoding whether arguments are conflicting have been proposed (Chen et al., 2009; Chen and Ji, 2009; Chen and Ng, 2016). More sophisticated features based on different kinds of similarity measures have also been considered, such as the surface similarity based on Dice coefficient and the WuPalmer WordNet similarity between argument heads (McConky et al., 2012; Cybulska and Vossen, 2013; Araki et al., 2014; Krause et al., 2016). These features are computed using either the outputs of event argument extractors and entity coreference resolvers (Ahn, 2006; Chen and Ng, 2014, 2015a; Lu and Ng, 2016) or the outputs of semantic parsers (Bejan and Harabagiu, 2014; Yang et al., 2015; Peng et al., 2016), and therefore suffer from error propagation (see Lu and Ng (2018)). Several previous works proposed joint models to address this problem (Lee et al., 2012; Lu et al., 2016), while others utilized iterative methods to propagate argument information (Liu et al., 2014; Choubey and Huang, 2017) in order to alleviate this issue. Nevertheless, argument extraction remains a very challenging task, especially when the arguments do not appear in the same sentence as the trigger. Our discourse-based pruning method can be thought of as a way of approximating argument compatibility without performing argument extraction.

6 Conclusion

We incorporated non-local information into a state-of-the-art joint model for event coreference resolution via topic modeling and discourse-based pruning. The resulting model not only significantly outperforms the independent models but also achieves the best results to date on the KBP 2017 English and Chinese event coreference corpora.

Acknowledgments

We thank the three anonymous reviewers for their detailed and insightful comments on an earlier draft of the paper. This work was supported in part by NSF Grants IIS-1528037 and CCF-1848608.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the COLING/ACL Workshop on Annotating and Reasoning about Time and Events*, pages 1–8.

Jun Araki, Zhengzhong Liu, Eduard Hovy, and Teruko Mitamura. 2014. Detecting subevent structure for event coreference resolution. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 4553–4558.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the LREC Workshop on Linguistic Coreference*, pages 563–566.

Cosmin Adrian Bejan and Sanda Harabagiu. 2014. Unsupervised event coreference resolution. *Computational Linguistics*, 40(2):311–347.

Chen Chen and Vincent Ng. 2014. SinoCoreferencer: An end-to-end Chinese event coreference resolver. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 4532–4538.

Chen Chen and Vincent Ng. 2015a. Chinese event coreference resolution: An unsupervised probabilistic model rivaling supervised resolvers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1097–1107.

Chen Chen and Vincent Ng. 2015b. Chinese zero pronoun resolution: A joint unsupervised discourse-aware model rivaling state-of-the-art resolvers. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 320–326.

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 30th AAAI Conference on Artificial Intelligence*, pages 2913–2920.

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.

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 International Workshop on Events in Emerging Text Types*, pages 17–22.

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.

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*, pages 156–163.

Agata Cybulska and Piek Vossen. 2015. Translating granularity of event slots into features for event coreference resolution. In *Proceedings of the 3rd Workshop on EVENTS*, pages 1–10.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

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*, pages 1971–1982.

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.

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

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.

Zhengzhong Liu, Jun Araki, Eduard Hovy, and Teruko Mitamura. 2014. Supervised within-document event coreference using information propagation. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 4539–4544.

Jing Lu and Vincent Ng. 2016. Event coreference resolution with multi-pass sieves. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*.

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.

Jing Lu and Vincent Ng. 2017b. UTD's event nugget detection and coreference system at KBP 2017. In *Proceedings of the 2017 Text Analysis Conference*.

Jing Lu and Vincent Ng. 2018. Event coreference resolution: A survey of two decades of research. In *Proceedings of the 27th 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 the 26th International Conference on Computational Linguistics*, pages 3264–3275.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Andrew Kachites McCallum. 2002. MALLET: A machine learning for language toolkit. http://www.cs.umass.edu/~mccallum/mallet.

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

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309.

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.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. 2009. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 248–256.

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

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the Sixth Message Understanding Conference*, pages 45–52.

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.