New Insights into Cross-Document Event Coreference: Systematic Comparison and a Simplified Approach

Andres Cremisini & Mark A. Finlayson
School of Computing and Information Sciences
Florida International University
11200 S.W. 8th Street, CASE Building, Room 362, Miami, FL 33199
{acrem003,markaf}@fiu.edu

Abstract

Cross-Document Event Coreference (CDEC) is the task of finding coreference relationships between events in separate documents, most commonly assessed using the Event Coreference Bank+ corpus (ECB+). At least two different approaches have been proposed for CDEC on ECB+ that use only event *triggers*, and at least four have been proposed that use both *triggers* and *entities*. Comparing these approaches is complicated by variation in the systems' use of gold vs. computed labels, as well as variation in the document clustering pre-processing step. We present an approach that matches or slightly beats state-of-the-art performance on CDEC over ECB+ with only event trigger annotations, but with a significantly simpler framework and much smaller feature set relative to prior work. This study allows us to directly compare with prior systems and draw conclusions about the effectiveness of various strategies. Additionally, we provide the first cross-validated evaluation on the ECB+ dataset; the first explicit evaluation of the pairwise event coreference classification step; and the first quantification of the effect of document clustering on system performance. The last in particular reveals that while document clustering is a crucial pre-processing step, improvements can at most provide for a 3 point improvement in CDEC performance, though this might be attributable to ease of document clustering on ECB+.

1 Introduction

Cross-Document Event Coreference (CDEC) is a clustering problem with a seemingly straightforward objective: Assign every event mention in a corpus to exactly one set in which every mention in the set refers to the same real-world event. CDEC is often contrasted with *Within-Document Event Coreference* (WDEC), where all the event mentions are drawn from the same document. All systems previously described in the literature approach CDEC in two steps: first, grouping documents into topical clusters (*document clustering*), followed by grouping events within each document cluster (*event clustering*). The CDEC literature defines events (probably incompletely) as linguistic objects comprised of a *trigger* and a set of *arguments*. The trigger is the word or phrase (usually a verb, though also commonly a noun phrase) that most closely describes the event, and the arguments are modifiers that would distinguish two events with identical triggers. Arguments are always entities, including things like times, locations, and human or non-human participants.

For example, consider the statements *"Yanitza went for a run"* and *"Juan went for a run,"* describing two distinct events. Note that names of the human participants, *Yanitza* and *Juan*, are arguments that distinguish otherwise identical events. The events often also have internal structure: the event trigger contains a light verb construction using *went* in combination with *run*. Add the complexity that these event mentions might be found in completely different contexts from completely different documents, and this simple example illustrates why event annotation—and the related task of event coreference, cross-document or not—is difficult and prone to error.

In seeking to build a CDEC system for our own use, we began with a thorough review of prior work. We discovered that prior systems were not well compared or evaluated, and that the performance of the key step of document clustering was often not reported. On the basis of these insights, we developed a system with a focus on simplicity and explainability. We identify issues in the CDEC literature that make comparing prior work difficult

The data and code for the experiments described herein is available at https://doi.org/10.34703/gzx1-9v95/FQVNQY.

Proceedings of the 1st Joint Workshop on Narrative Understanding, Storylines, and Events, pages 1–10
July 9, 2020. ©2020 Association for Computational Linguistics

and suggest best practices to remedy this situation going forward. Our system is modeled on the BAG OF EVENTS system described by Vossen and Cybulska (2018), primarily because of its simplicity and strong performance. However, we use a different and significantly smaller feature set to predict pairwise event coreference (4 features instead of 19), we employ a different document clustering scheme independent of gold-standard annotations, we ingest only event trigger annotations (instead of both triggers and entities), and we develop a different event clustering technique while maintaining comparable state-of-the-art performance.

The paper proceeds as follows. We begin with an extensive review of the area: the two major corpora, prior work in CDEC, as well as some relevant WDEC work (§2). We then describe our own, simplified approach, with careful attention to evaluating all stages of the pipeline (§3). We discuss our cross-validated results for various scenarios (§4) and conclude with a list of contributions (§5).

2 Prior Work

2.1 Data: The ECB & ECB+ Corpora

Most CDEC systems have been developed and evaluated on the EventCorefBank (ECB) and EventCorefBank+ (ECB+) corpora, with most using ECB+ because it is larger. ECB was the first corpus developed specifically for CDEC (Bejan and Harabagiu, 2010). It comprises 482 documents selected from GoogleNews, clustered into 43 topics, with each topic containing documents on a specific event, such as the 2009 Indonesian earthquake or the 2008 riots in Greece over a teenager's death. The corpus is annotated using a "bag of events" and entities approach, where co-referring events are all placed into the same group along with their related entities, but relationships between specific entities and events are not recorded. A limitation of this annotation scheme is that it makes it impossible to differentiate events based on their arguments.

ECB+ extends ECB with 500 articles (bringing the total to 982) that refer to similar but unrelated events across the same 43 topics (Cybulska and Vossen, 2014). For example, the topic with the 2009 Indonesian earthquake was expanded with texts referring to the 2013 Indonesian earthquake. These extra texts were marked with a different subtopic. In the release notes of ECB+, the authors recommend using a subset of 1,840 sentences that were additionally checked for correctness of coreference annotations. We restrict our experiments to these double-checked sentences; they contain 5,726 events and 897 coreference chains with an average length of 5.5 events per chain ($\sigma = 6.1$). Figure 1 shows a boxplot of chain lengths, which shows that most coreference chains in the data are quite short, with only a handful (around 20) greater than 15 events in length.

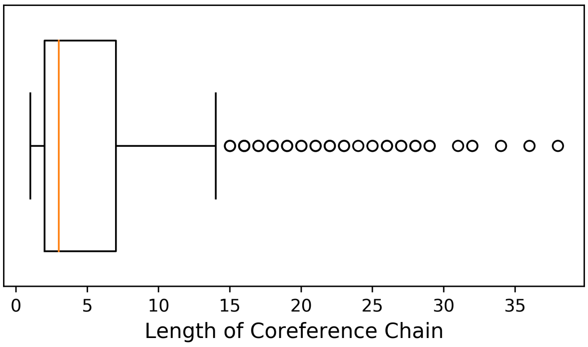

Figure 1: Boxplot of Coreference Chain Lengths in ECB+

Other datasets are available for the WDEC task only, namely KBP and ACE (Getman et al., 2018; Doddington et al., 2004). These datasets employ a different, richer event annotation than ECB/+, including types of events and temporal relationships between events. However, these corpora provide only WDEC annotations (and are also not free, being distributed by the Linguistic Data Consortium).

All extant CDEC systems begin with document clustering followed by event clustering, either by computing document clusters or using gold-standard topic or sub-topic labels. Most CDEC systems approach document clustering with off-the-shelf algorithms, and in the experimental setups used with the ECB+ corpus these algorithms tend to work quite well, though we discuss some subtleties in Section 4.2. All approaches make use of event trigger or event trigger combined with entity information, either gold-standard or computed.

2.2 Early Approaches

Early CDEC resolution systems used approaches that have not been carried into more recent work. Bejan and Harabagiu (2010) used a Bayesian approach that used a Dirichlet Process with a Chinese Restaurant prior to find the configuration of event clusters with greatest probability given the data. They used gold-standard document clusters, but did not make use of gold-standard event annotations, rather using an event and entity extractor developed in earlier work and augmenting the predicted events using a semantic parser. They tested their model on the ECB dataset, and achieved an overall performance of 0.52 CoNLL F_1. Notably, this is the only system in prior work that reports cross-validation results, but they did not report the performance of their event detection system.

Chen and Ji (2009), in contrast, developed an approach that formulates *Within-Document* Event Coreference (WDEC) as a spectral graph clustering problem. Although this system was tested on the ACE dataset, which only includes WDEC annotations (not CDEC), its performance of 0.836 Constrained Entity-Alignment F-measure (Luo, 2005) (CEAF, a.k.a., the ECM, or Entity Constrained-Mention, F-measure) is of interest to work on CDEC.

2.3 Later Approaches

All recent CDEC systems divide into *event-only* clustering and *joint event-entity* clustering. Event-only systems only perform event clustering (though some use entity information to augment their feature sets) while joint event-entity systems resolve event and entity coreference simultaneously.

2.3.1 Event-Only Clustering

Kenyon-Dean et al. (2018) describe an event-only clustering approach that generates event embeddings for clustering within the hidden layer of a neural network. The paper does not specify if document clustering was performed before CDEC, or if they used gold-standard labels. Using only event trigger annotations, the authors trained a neural network with a single hidden layer to predict the event cluster of an event given its feature representation (e.g. *word2vec* embeddings). Since their interest was clustering and not classification, however, they constrained the training loss function in such a way as to produce more clusterable event embeddings in the model's hidden layer. As a final step, they use the event embeddings of test set events as input to an agglomerative clustering algorithm.

Vossen and Cybulska (2018) describe two event-only systems, NEWSREADER and BAG OF EVENTS. The NEWSREADER system is a pipeline designed to track events in the news, with extensive use of rule-based components as well as machine-learning-based components. The *Bag of Events* system is a simpler, event-only clustering approach that achieves strong performance on ECB+; because of this we chose it as the starting point for our system, and as such we describe it in greater detail than other prior work. BAG OF EVENTS is based on a pairwise decision tree classifier trained at both the document and event level. The document-level classifier is trained to predict if two documents contain at least one pair of coreferring events, and the event-level classifier is trained to predict if two events corefer. The first step of BAG OF EVENTS is to run the document-level classifier on every pair of documents in the test set, placing those documents that are predicted as coreferent together in the same set. Once documents are clustered, the event-level classifier is run on every pair of events in the cluster, followed by computing the transitive closure to find the final event clusters.

Both the document-level and event-level classifiers use the same features, but are computed at different levels of granularity by comparing a pair of document or event "templates." A *template* is defined by the "bag of events" principle, where each event is represented as a collection of slots (action, time, location, etc., see Table 1) where each slot contains the union of items that fill slot across all event mentions in the relevant unit of discourse. A document template's unit of discourse is the document itself, and an event's unit of discourse are the sentence where it appears. For example, if we take the two sentences in Example (1) as a document, we can derive the document and event templates as shown in Table 1.

(1) *The "American Pie" actress has entered Promises for undisclosed reasons. The actress, 33, reportedly headed to a Malibu treatment facility on Tuesday.*

The feature vector for a pair of templates is derived by computing 19 overlap features between the corresponding slots of each template. 5 features are derived from event triggers, and the remaining 14 from entities. This approach is attractive because of its conceptual uniformity and simplicity, essentially repeating the same step at two levels of granularity. The drawbacks are a large feature set, dependence on both trigger and entity annotations, and an extremely simple clustering procedure; we designed our system to address these issues.

2.4 Joint Event-Entity Clustering

In contrast to event-only clustering, joint event-entity clustering attempts to resolve event and entity coreference simultaneously, using information

		Templates	
Slot	Event 1	Event 2	Document
Action	*entered*	*headed*	*entered, headed*
Time	-	*on Tuesday*	*on Tuesday*
Location	*Promises*	*Malibu treatment facility*	*Promises, Malibu treatment facility*
Human Participant	*actress*	*actress*	*actress*
non-Human Participant	-	-	-

Table 1: Event and Document Templates in Example (1)

from either step to inform the decisions made by the other. Lu and Ng (2017) described a system that jointly learns event triggers, anaphoric event relationships, and non-anaphoric event coreference relationships. They only perform *Within-Document* Event Coreference (WDEC) and evaluate their model on the KBP 2016 English and Chinese datasets for event coreference. Their formulation makes explicit use of discourse information within the document to construct a conditional random field (CRF) that performs the classification. Given the conceptual differences between KBP 2016 and ECB+ it is difficult to compare results across the two datasets. However, Lu and Ng (2017) reported state-of-the-art performance on KBP 2016 at the time.

Lee et al. (2012) described a system that computes event triggers and entities using a publicly available system that performs nominal, pronominal, and verbal mention extraction. After extracting all candidate event or entity mentions, they make use of a publicly available WDEC resolution system that applies a series of high precision deterministic rules to decide coreference. Using this initial clustering, they trained a linear regressor that predicts the quality of merging two clusters (where quality is defined as the number of correct pairwise links divided by the number of total pairwise links), merging clusters in decreasing order of predicted quality. They did not distinguish between events and entities at clustering time, but rather perform cluster merges using features derived from the relationships between the mentions in two candidate clusters, relying heavily on a semantic role labeler (SRL). They use the ECB dataset, adding a series of event and entity coreference annotations.

Barhom et al. (2019) describe a system inspired by Lee et al. (2012), developed on ECB+. The system performs document clustering using K-means and then uses gold-standard event trigger and entity annotations to generate vector embeddings for events and entities, including both character-, word-, and context-embeddings (ELMo is used for the context embeddings; Peters et al., 2018). Together with these vectors the system uses a *dependency vector*, which is the concatenation of a set of vectors designed to capture inter-dependency between event and entity mentions. For entities, this set includes an embedding for the event head that the entity modifies as well as the embeddings for the event heads of all coreferring events. For events, the set includes entity embeddings for each of four event roles (ARG0, ARG1, TMP, LOC) that combine the embedding for the modifying entity mention and the embeddings of all other entity mentions that corefer with the modifying entity. The system computes event and entity clusters iteratively, recomputing the dependency vectors as clusters are merged. They employ an agglomerative clustering algorithm furnished with two trained pairwise prediction functions that output the likelihood that two pairs of events or entities corefer.

3 Simplified Approach

We based our approach on the Bag of Events system described by Vossen and Cybulska (2018) and discussed above in Section 2.3, primarily because of its simplicity and strong performance. However, we made several modifications based on what we learned in our literature survey:

- We use a different and significantly smaller feature set to predict pairwise event coreference (4 features instead of 19);
- We employ a different document clustering scheme independent of gold-standard annotations;
- We ingest only event trigger annotations, instead of both triggers and entities; and,
- We developed a different event clustering tech-

nique.

These modifications simplified the approach while maintaining comparable performance. At a high level, our pipeline first performs document clustering and then uses a trained pairwise event coreference classifier as the essential component in an event clustering procedure that generates CDEC chains.

3.1 Document Clustering

Like all extant systems, we first perform document clustering to assemble clusters within which event coreference will be performed. We represent our documents as a bag-of-words vector with *tf-idf* weights and perform clustering using affinity propagation (Frey and Dueck, 2007) with the damping parameter set to 0.5. On the test set used by Vossen and Cybulska $(2018)_{BoE}$ we achieve near perfect document clustering performance, as detailed in Section 4.2. This strong document clustering performance is reported by other researchers as well (Barhom et al., 2019; Choubey and Huang, 2017); Vossen and Cybulska $(2018)_{BoE}$ do not provide these numbers. The document clustering step, employed in some form by all CDEC systems, is essentially a high recall, low precision class balancing scheme that significantly reduces the number of false event coreference pairs while retaining a high percentage of true coreference pairs. This reduces the search space of event pairs before building CDEC chains and makes it easier to train a classifier with a more balanced training set.

3.2 Pairwise Event Coreference Classifier

The training data for our pairwise event coreference classifier comprises all possible event pairs within a gold-standard ECB+ sub-topic document cluster, labeled as either coreferring or not. We use a shallow, fully-connected neural network with one hidden layer composed of two nodes to predict coreference between two events. We choose this classifier because neural networks of this sort are adept at modeling the class probability of a prediction, which is a feature we make use of in our event clustering scheme by picking a cutoff for true predictions (Scikit-Learn, 2019). We tried a number of other classifier types (e.g., RDF, SVM, regression, more complex MLP architectures), but they all equivalent or worse performance. After training the classifier we use a held-out development set (20% of the training samples) to perform grid search to find a confidence threshold that maximizes the classifier's F_β score. The value of β we used and the reasoning behind our choice is detailed in Section 4.4. Note that at testing time, we use computed document clusters to generate the dataset of event pairs, inevitably losing some corefering event pairs that are erroneously placed in different document clusters. The classifier uses four features, listed below, to predict pairwise event coreference.

Feature 1: Head Phrase Word Similarity (Vec) This feature captures the semantic similarity of two events by measuring the average cosine similarity of each word in two events' triggers using pre-trained Fasttext word embeddings (Bojanowski et al., 2016). Our experiments (shown below) indicate that this feature accounts for the majority of the performance of the pairwise classifier.

Feature 2: Event Word Distribution (WD) This feature captures the lexical similarity between the overall textual expression of the event, including modifiers and slot fillers. Starting from the gold-standard trigger annotations provided by ECB+, we identify the *event text*—the set of words related to each event—by collecting all of the event's trigger words and their dependent words as found in the dependency graph of the sentence (we computed the dependency graphs using Stanford CoreNLP; Manning et al., 2014). For both events we construct a vector where each element represents a surface form found in the union of both sentences, and the value of each cell is the term frequency of that form: the number of tokens of that form found in the event text, divided the total number of tokens across both sentences. We found that term frequency worked better than a *tf-idf* type measure. The feature itself is the cosine similarity between the two vectors. This is the second most useful feature.

Feature 3: Relative Sentence Similarity (SS) Whereas the event word distribution feature is meant to capture the relative lexical similarity of events themselves, relative sentence similarity is designed to capture the relative lexical similarity of their sentence contexts. The sentences in each event's document are treated themselves as documents in order to compute a *tf-idf* vector for each event's sentence. For example, if two events appear in the same sentence their *tf-idf* vectors are identical. As for the event word distribution feature, the relative sentence similarity feature itself is the

	B^3			$CEAF_e$			MUC			$CoNLL$
Feature Set	P	R	F_1	P	R	F_1	P	R	F_1	F_1
POS only	99.9	18.2	30.7	9.6	51.5	16.1	20.0	0.1	0.2	15.6
SS only	23.7	82.7	36.7	25.6	13.3	17.5	74.5	83.4	78.7	45.9
WD only	38.3	75.5	50.6	36.2	32.0	33.8	81.9	83.8	82.8	55.7
VEC only	78.6	74.4	76.1	52.2	68.9	59.2	92.2	85.4	88.7	74.6
All except Vec	61.4	66.0	63.2	37.4	57.6	45.3	87.6	76.9	81.8	63.4
All except SS	80.9	72.6	76	50.3	70.5	58.5	92.7	84.2	88.2	74.3
All except POS	80.6	73.0	76.2	51.2	70.1	59.1	92.4	84.4	88.2	74.5
All except WD	80.5	73.3	76.4	51.4	70.7	59.3	92.5	84.8	88.4	74.7
All (Vec, WD, SS, POS)	**82.2**	**72.5**	**76.8**	**51.3**	**71.2**	**59.4**	**92.5**	**84.5**	**88.3**	**74.8**

Table 2: Feature Ablation Study on CDEC Performance (5-fold CV), using Gold triggers and Gold document clusters.

cosine similarity between the vectors of the two sentences. This is the third most useful feature.

Feature 4: Head Phrase Part of Speech (POS) This is a binary feature that is assigned a value of 1 if two events' triggers have the same part of speech (*noun*, *verb*, or *other*) and a 0 if they differ. This is the least useful feature.

3.3 Event Clustering

Final event clustering relies on the pairwise event classifier prediction confidence. First, we use the pairwise event classifier to predict a coreference confidence for all event pairs in the set and rank the pairs in decreasing order of classifier confidence. Confidences above a certain the cutoff are clustered using transitive closure. We chose the cutoff to maximize an intermediate measure, F_β, where β is chosen by tuning on the development set. All events not assigned to a cluster in this step were assigned to singleton clusters. We attempted to use affinity propagation as a clustering scheme with our trained classifier as a distance function, but this performed significantly worse. Nevertheless, one drawback of relying on pairwise distances (rather than embedding in a metric space) for clustering is that we cannot use clustering algorithms that perform vector arithmetic between single instances, significantly limiting our design choices.

The relative contributions of the different features to the overall performance is shown in Table 2. We performed this ablation study with gold event triggers and gold document clusters.

4 Results

4.1 CDEC

Table 3 shows results for all combinations of gold and computed labels using 5-fold cross validation. We use 5-fold cross validation because it generates test sets of roughly the same size as a commonly used test set amongst systems that use ECB+ (topics 36-45). Ours is the first study to report cross-validated results on ECB+, though we report our system's performance on two different test sets in Section 4.6 in order to compare with prior work.

4.2 Document Clustering

Our experiments show that on average, document clustering on ECB+ is responsible for about 3 CoNLL F_1 points, as shown in the difference between rows 1 and 2 in Table 3. Despite this modest performance loss, there is cause to doubt that this generalizes to document collections "in the wild," since ECB+ document clusters correspond to fairly distinct events with little lexical overlap that are probably relatively easy to cluster. In any case, document clustering is an important step for CDEC resolution. Without document clustering, the testing false/true ratio on ECB+ over 5 cross-validation folds is 89:1 (544,157 false pairs and 6,113 true pairs) on average. With document clustering, the false/true ratio drops to 6:1 (26,416 false pairs and 4,836 true pairs); the cost is that we lose some corefering event pairs—13% on average—but we gain a procedure with a tractable running time and higher performance. Details of the clustering performance are shown in Table 4.

Doc.	*Ev.*	B^3			$CEAF_e$			MUC			$CoNLL$
Clust.	*Trig.*	P	R	F_1	P	R	F_1	P	R	F_1	F_1
Gold	Gold	82.2	72.5	76.8	51.3	71.2	59.4	92.5	84.5	88.3	**74.8**
Pred.	Gold	78.5	68.4	72.8	48.3	68.8	56.6	90.5	81.8	85.8	71.7
Gold	Pred.	44.3	26.2	32.9	18.8	37.0	24.8	64.3	34.3	44.6	34.1
Pred.	Pred.	45.2	24.6	31.7	18.1	37.9	24.4	64.0	32.4	42.8	33.0

Table 3: CDEC Performance (5-fold CV)

ARI	*V-Measure*	*Homogeneity*	*Completeness*
0.85	0.94	0.97	0.91

Table 4: Document Clustering Performance (5-fold CV)

4.3 Computed Event Triggers

The most striking performance drop occurs when we remove gold-standard event triggers, showing that trigger detection is a major performance bottleneck for CDEC, responsible for about 40 CoNLL F_1 points on average. To detect triggers we use the freely available pre-trained CAEVO Event Trigger extraction system (Chambers et al., 2014), which achieves modest performance on ECB+ of 0.62 precision, 0.43 recall, and 0.51 F_1. The CAEVO system achieved state-of-the-art performance at time of publication, and was in our experience the simplest event extraction system to integrate.

4.4 Pairwise Event Coreference Classifier

Using a cutoff of 0.72, the pairwise event classifier achieved a maximum *in vitro* performance (that is, in isolation from the rest of the system) of 0.64 precision, 0.55 recall, 0.59 F_1, and 0.95 accuracy. The cutoff is the confidence level above which a pairwise event coreference judgement is retained. We tuned the cutoff on the development set[1].

4.5 Feature Analysis

We perform logistic regression on the entire ECB+ dataset in order to investigate the predictive power of our feature set. While we do not use logistic regression as our classifier, given that our shallow neural network is a concatenation of gated logistic regressions trained by minimizing overall classification error, analysis of logistic regression provides useful insight into our feature set.

Feature	Coef.	Std. Err.	p-value
Vec	7.12	0.068	0.000
WD	0.89	0.072	0.000
SS	-0.50	0.069	0.000
POS	-0.31	0.045	0.000
constant	-3.18	0.045	0.000

Table 5: Logistic Regression Coefficients (all ECB+). The p-value is computed for $\alpha = 0.05$.

The regression coefficients in Table 5 clearly show that the most powerful feature is the word vector feature (Vec), the word embedding head phrase similarity. In fact, training a simple logistic regression with only an intercept and the word vector feature gives a 5-fold cross-validated CoNLL F_1 of 70.7 and 69.2 on topics 36-45.

4.6 Comparison with Prior Work

Comparing the performance of existing ECB+ CDEC systems is unfortunately quite difficult due to a wide variation in testing schemes and usage of gold-standard annotations. Because of this, it is not possible to clearly determine which system achieves state-of-the-art performance. In an attempt to provide a fair comparison amongst existing systems, Table 6 shows performance of all prior work evaluated on ECB+ grouped by test sets and gold-standard annotations. Minding these conditions, we can currently only determine state-of-the-art performance on a given test set using a given set of gold-standard annotations.

4.6.1 Test Sets and Gold-Standard Annotations

Unfortunately, none of the existing CDEC papers provide a reasoning behind their choice of test set; in fact, the choices seem quite arbitrary. Standard practice in NLP suggests that multi-fold cross val-

[1] An interesting aside is that, through additional experimentation we found that if one wished to tune the pairwise event coreference classifier in isolation to maximize CDEC performance, the appropriate metric to maximize is $F_{0.8}$ for the pairwise classifier, rather than F_1.

			B^3			$CEAF_e$			MUC			$CoNLL$
	Gold	Test	P	R	F_1	P	R	F_1	P	R	F_1	F_1
OURS	T	36–45	74.3	69.2	**71.6**	49.6	60.7	54.6	89.4	84.9	**87.1**	**71.1**
KD2018			71	67	69	71	67	**69**	67	71	69	69
OURS	T	24–43	73.6	65.8	**69.5**	40.8	60.3	48.7	88.7	80.7	**84.5**	**67.6**
CH2017			56.2	66.6	61	59	54.2	**56.5**	67.5	80.4	73.4	63.6
Bh2019	T+E	36–45	76.1	85.1	**80.3**	81	73.8	**77.3**	77.6	84.5	**80.9**	**79.1**
CV2018 (BoE)			71	78	74	-	-	64	71	75	73	73
CV2018 (NwR)	T+E	24–43	72.8	64.2	**68.3**	55	65.4	**59.7**	77.4	69.7	**73.3**	**67.1**
YC2015*	-	24–43	78.5	40.6	53.5	38.6	68.9	49.5	80.3	67.1	73.1	**58.7**

Table 6: CDEC Performance on Single Test Set. KD2018 = Kenyon-Dean et al. (2018); CH2017 = Choubey and Huang (2017); Bh2019 = Barhom et al. (2019); CV2018 = Vossen and Cybulska (2018); YC2015 = Yang et al. (2015). *YC2015 computes event triggers and entities

idation (CV) should clearly be used. In our experiments, we used 5-fold CV, after noting that our system performed similarly using 10-fold cross validation as well as with 10 runs of randomized 5-fold and 10-fold cross validation, respectively. 5-fold cross validation is also useful for comparison with published systems because it generates test sets of roughly the same size as the previously used test set of topics 36-45. Using 2-fold cross validation to approximate the size of test set 24-43 (the other previously used test set) seems less useful.

Comparing the performance between trigger-only CDEC systems and CDEC systems that use triggers & entities is more difficult. Computing entities as well as event triggers adds an additional potential source of error, and if researchers did not report evaluation of their entity extraction systems independently of the rest of the pipeline, the contribution of those components cannot be separated from the whole. Current state-of-the-art entity detection systems perform at around 0.90 F_1 on the OntoNotes 5.0 corpus (Strubell et al., 2017), whereas state-of-the-art trigger detection systems perform at around 0.80 F_1 on the ACE2005 dataset (Yang et al., 2019). Of course, finding implementations of state-of-the-art systems or implementing them from scratch is a task onto itself. There is currently no evaluation of trigger or entity detection performance on the entire ECB+ dataset. Yang et al. (2015) describe the only system that makes exclusive use of computed trigger and entity labels on ECB+. They report that their trigger and entity detection system correctly identifies 95% of actions, 90% of participants, 94% of times and 74% of locations, but these results apply only to a development set comprised of topics 21-23; they do not provide the system's performance on any other subset of ECB+. Despite these difficulties, the results of Barhom et al. (2019) do seem to suggest that adding in entities results in a substantial improvement in performance.

4.6.2 Document Clustering

Reporting of the source of document cluster labels is inconsistent across the literature. Yang et al. (2015) is the the only ECB+ system that does not use document clustering as a pre-processing step, instead using gold-standard labels to restrict the search space for CDEC. Kenyon-Dean et al. (2018) do not specify if they use computed or gold-standard document clusters. We believe it is reasonable to separate document clustering performance from CDEC performance—events and documents are fairly distinct objects with different structures that require different techniques to determine their similarity. Practically, however, it seems that document clustering is a necessary pre-processing step in order to make CDEC tractable, as outlined in Section 4.2. For these reasons, we suggest that future CDEC systems report on performance both with and without gold-standard document clusters.

5 Contributions

We have presented a simple, event-trigger-only CDEC system that achieves strong performance on ECB+ compared with other trigger-only CDEC systems. We have compared our approach, where possible, with prior work and highlighted the difficulties in comparing existing ECB+ systems, providing suggestions for evaluation criteria in future work. We presented performance results of all components of our pipeline and quantified how error on each component propagates to downstream CDEC performance. We also provided cross validated results, the first ECB+ CDEC study to do so.

Acknowledgments

This work was supported by Office of Naval Research (ONR) grant number N00014-17-1-2983.

References

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. *arXiv*, 2.

Cosmin Bejan and Sanda Harabagiu. 2010. Unsupervised Event Coreference Resolution with Rich Linguistic Features. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1412–1422.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense Event Ordering with a Multi-Pass Architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Zheng Chen and H Ji. 2009. Graph-based event coreference resolution. *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 54–57.

Prafulla Kumar Choubey and Ruihong Huang. 2017. Event Coreference Resolution by Iteratively Unfolding Inter-dependencies among Events. *arXiv*.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4545–4552.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. *Proceedings of LREC*, 2.

Brendan J. Frey and Delbert Dueck. 2007. Clustering by passing messages between data points. *Science*, 315(5814):972–976.

Jeremy Getman, Joe Ellis, Stephanie Strassel, Zhiyi Song, and Jennifer Tracey. 2018. Laying the groundwork for knowledge base population: Nine years of linguistic resources for TAC KBP. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Kian Kenyon-Dean, Jackie Chi Kit Cheung, and Doina Precup. 2018. Resolving Event Coreference with Supervised Representation Learning and Clustering-Oriented Regularization. *arXiv*.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint Entity and Event Coreference Resolution across Documents. *(EMNLP-CoNLL 2012) Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500.

Jing Lu and Vincent Ng. 2017. Joint Learning for Event Coreference Resolution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 90–101.

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, Canada.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The {Stanford} {CoreNLP} Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Scikit-Learn. 2019. Scikit-learn probability calibration. `https://scikit-learn.org/stable/modules/calibration.html`. Accessed: 2019-11-23.

Emma Strubell, Pat Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *EMNLP*.

Piek Vossen and Agata Cybulska. 2018. Identity and Granularity of Events in Text. In *Lecture Notes in Computer Science*, volume 9624 LNCS, pages 501–522. Springer-Verlag.

Bishan Yang, Claire Cardie, and Peter Frazier. 2015. A Hierarchical Distance-dependent Bayesian Model for Event Coreference Resolution. *arXiv*.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring Pre-trained Language Models for Event Extraction and Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, Florence, Italy. Association for Computational Linguistics.