Revisiting Joint Modeling of Cross-document Entity and Event Coreference Resolution

Shany Barhom[1], **Vered Shwartz**[1], **Alon Eirew**[2], **Michael Bugert**[3], **Nils Reimers**[3], and **Ido Dagan**[1]

[1] Computer Science Department, Bar-Ilan University
[2] Intel AI Lab, Israel
[3] Ubiquitous Knowledge Processing Lab, Technische Universitat Darmstadt, Germany
{shanyb21,vered1986}@gmail.com, alon.eirew@intel.com
{bugert,reimers}@ukp.informatik.tu-darmstadt.de, dagan@cs.biu.ac.il

Abstract

Recognizing coreferring events and entities *across* multiple texts is crucial for many NLP applications. Despite the task's importance, research focus was given mostly to within-document entity coreference, with rather little attention to the other variants. We propose a neural architecture for cross-document coreference resolution. Inspired by Lee et al. (2012), we jointly model entity and event coreference. We represent an event (entity) mention using its lexical span, surrounding context, and relation to entity (event) mentions via predicate-arguments structures. Our model outperforms the previous state-of-the-art event coreference model on ECB+, while providing the first entity coreference results on this corpus. Our analysis confirms that all our representation elements, including the mention span itself, its context, and the relation to other mentions contribute to the model's success.

1 Introduction

Recognizing that various textual spans across multiple texts refer to the same entity or event is an important NLP task. For example, consider the following news headlines:

1. ***2018 Nobel prize for physics*** *goes to* ***Donna Strickland***
2. ***Prof. Strickland*** *is awarded* ***the Nobel prize for physics***

Both sentences refer to the same entities (Donna Strickland and the Nobel prize for physics) and the same event (awarding the prize), using different words. In coreference resolution, the goal is to cluster expressions that refer to the same entity or event in a text, whether within a single document or across a document collection. Recently, there has been increasing interest in cross-text inferences, for example in question answering (Welbl et al., 2018; Yang et al., 2018; Khashabi et al., 2018; Postma et al., 2018). Such applications would benefit from effective cross-document coreference resolution.

Despite the importance of the task, the focus of most coreference resolution research has been on its within-document variant, and rather little on cross-document coreference (CDCR). The latter is sometimes addressed partially using entity linking, which links mentions of an entity to its knowledge base entry. However, cross-document entity coreference is substantially broader than entity linking, addressing also mentions of common nouns and unfamiliar named entities.

The commonly used dataset for CDCR is ECB+ (Cybulska and Vossen, 2014), which annotates within-document coreference as well. The annotations are denoted separately for entities and events, making it possible to solve one task while ignoring the other. Indeed, to the best of our knowledge, all previously published work on ECB+ addressed only event coreference.

Cross-document entity coreference has been addressed on EECB, a predecessor of the ECB+ dataset. Lee et al. (2012) proposed to model the entity and event coreference tasks jointly, leading to improved performance on both tasks. Their model preferred to cluster event mentions whose arguments are in the same entity coreference cluster, and vice versa. For instance, in the example sentences above, a system focusing solely on event coreference may find it difficult to recognize that *goes to* and *awarded* are coreferring, while a joint model would leverage the coreference between their arguments.

Inspired by the success of the joint approach of Lee et al. (2012), we propose a joint neural architecture for CDCR. In our joint model, an event (entity) mention representation is aware of other entities (events) that are related to it by predicate-argument structure. We cluster mentions based on a learned pairwise mention coreference scorer.

A disjoint variant of our model, on its own, improves upon the previous state-of-the-art for event

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4179–4189
Florence, Italy, July 28 - August 2, 2019.

coreference on the ECB+ corpus (Kenyon-Dean et al., 2018) by 9.5 CoNLL F_1 points. To the best of our knowledge, we are the first to report performance on the entity coreference task in ECB+.

Our joint model further improves performance upon the disjoint model by 1.2 points for entities and 1 point for events (statistically significant with $p < 0.001$). Our analysis further shows that each of the mention representation components contributes to the model's performance.[1]

2 Background and Related Work

Coreference resolution is the task of clustering text spans that refer to the same entity or event. Variants of the task differ on two axes: (1) resolving entities ("*Duchess of Sussex*", "*Meghan Markle*", "*she*") vs. events ("*Nobel prize for physics [goes to] Donna Strickland*", "*Donna Strickland [is awarded] the 2018 Nobel prize for physics*"), and (2) whether coreferring mentions occur within a single document (WD: *within-document*) or across a document collection (CD: *cross-document*).

2.1 Datasets

The largest datasets that include WD and CD coreference annotations for both entities and events are EECB (Lee et al., 2012) and ECB+ (Cybulska and Vossen, 2014). Both are extensions of the Event Coreference Bank (ECB) (Bejan and Harabagiu, 2010) which consists of documents from Google News clustered into topics and annotated for event coreference. Entity coreference annotations were first added in EECB, covering both common nouns and named entities.

ECB+ increased the difficulty level by adding a second set of documents for each topic (subtopic), discussing a different event of the same type (*Tara Reid enters a rehab center* vs. *Lindsay Lohan enters a rehab center*). The annotation is not exhaustive, where only a number of salient events and entities in each topic are annotated.

2.2 Models

Entity Coreference. Of all the coreference resolution variants, the most well-studied is WD entity coreference resolution (e.g. Durrett and Klein, 2013; Clark and Manning, 2016). The current best performing model is a neural end-to-end system which considers all spans as potential entity mentions, and learns distributions over possible antecedents for each (Lee et al., 2017). CD entity coreference has received less attention (e.g. Bagga and Baldwin, 1998b; Rao et al., 2010; Dutta and Weikum, 2015), often addressing the narrower task of entity linking, which links mentions of known named entities to their corresponding knowledge base entries (Shen et al., 2015).

Event Coreference. Event coreference is considered a more difficult task, mostly due to the more complex structure of event mentions. While entity mentions are mostly noun phrases, event mentions may consist of a verbal predicate (*acquire*) or a nominalization (*acquisition*), where these are attached to arguments, including event participants and spatio-temporal information.

Early models employed lexical features (e.g. head lemma, WordNet synsets, word embedding similarity) as well as structural features (e.g. aligned arguments) to compute distances between event mentions and decide whether they belong to the same coreference cluster (e.g. Bejan and Harabagiu, 2010, 2014; Yang et al., 2015).

More recent work is based on neural networks. Choubey and Huang (2017) alternate between WD and CD clustering, each step relying on previous decisions. The decision to link two event mentions is made by the pairwise WD and CD scorers. Mention representations rely on pre-trained word embeddings, contextual information, and features related to the event's arguments.

Kenyon-Dean et al. (2018) similarly encode event mentions using lexical and contextual features. Differently from Choubey and Huang (2017), they do not cluster documents to topics as a pre-processing step. Instead, they encode the document as part of the mention representation.

Most of the recent models were trained and evaluated on the ECB+ corpus, addressing solely the event coreference aspect of the dataset.

Joint Modeling. Some of the prior models leverage the event arguments to improve their coreference decisions (Yang et al., 2015; Choubey and Huang, 2017), but mostly relying only on lexical similarity between arguments of candidate event mentions. A different approach was proposed by Lee et al. (2012), who jointly predicted event and entity coreference.

At the core of their model lies the assumption that arguments (i.e. entity mentions) play a key

[1]The code is available at https://github.com/shanybar/event_entity_coref_ecb_plus.

role in describing an event, therefore, knowing that two arguments are coreferring is useful for finding coreference relations between events, and vice versa. They incrementally merge entity or event clusters, computing the merge score between two clusters by learning a linear regression model based on discrete features.

Lee et al. (2012) evaluated their model on EECB, outperforming disjoint CD coreference models for both entities and events. Nonetheless, as opposed to the more recent models, their representations are sparse. Lexical features are based on lexical resources such as WordNet (Miller, 1995), which are limited in coverage, and context is modeled using semantic role dependencies, which often do not cover the entire sentential context. We revisit the joint modeling approach, trying to overcome prior limitations by using modern neural techniques, which provide better and more generalizable representations.

3 Model

We propose an iterative algorithm that alternates between interdependent entity and event clustering, incrementally constructing the final clustering configuration. A single iteration for events is as follows (entity clustering is symmetric). We start by computing the mention representations (Section 3.1), which couple the entity and event clustering processes. When predicting event clusters, the event mention representations are updated to consider the current configuration of entity clusters. The mention representations are then fed to an event mention pair scorer that predicts whether the mentions belong to the same cluster (Section 3.2). Finally, we apply agglomerative clustering where the cluster merging score is based on the predicted pairwise mention scores. Sections 3.3 and 3.4 detail the specifics of the inference and training procedures, respectively. Various implementation details are mentioned in Section 3.5.

3.1 Mention Representation

Given a mention m (entity or event), we compute a vector representation with the following features.

Span. We combine word-level and character-level features. We compute word-level representations using pre-trained word embeddings. For events, we take the embedding of the head word, while for entities we average over the mention's words. Character-level representations are complementary, and may help with out-of-vocabulary words and spelling variations. We compute them by encoding the span using a character-based LSTM (Hochreiter and Schmidhuber, 1997). The span vector $\vec{s}(m)$ is a concatenation of the word- and character-level vectors.

Context. The context surrounding a mention may indicate its compatibility with other candidate mentions (Clark and Manning, 2016; Lee et al., 2017; Kenyon-Dean et al., 2018). To model context, we use ELMo, contextual representations derived from a neural language model (Peters et al., 2018). ELMo has recently improved performance on several challenging NLP tasks, including within-document entity coreference resolution (Lee et al., 2018). We set the context vector $\vec{c}(m)$ to the contextual representation of m's head word, taking the average of the 3 ELMo layers.

Semantic dependency to other mentions. To model dependencies between event and entity clusters, we identify semantic role relationships between their mentions using a semantic role labeling (SRL) system.

For a given event mention m_{v_i}, we extract its arguments, focusing on 4 semantic roles of interest: `Arg0`, `Arg1`, `location`, and `time`. Consider a specific argument slot, e.g. `Arg1`. If the slot is filled with an entity mention m_{e_j} which in the current configuration is assigned to an entity cluster c, we set the corresponding `Arg1` vector to the averaged span vector of all the mentions in c: $\vec{d}_{\texttt{Arg1}}(m_{v_i}) = \frac{1}{|c|}\sum_{m \in c} \vec{s}(m)$. Otherwise we set $\vec{d}_{\texttt{Arg1}}(m_{v_i}) = \vec{0}$. The final vector $\vec{d}(m)$ is the concatenation of the various argument vectors:

$$\vec{d}(m_{v_i}) = [\vec{d}_{\texttt{Arg0}}(m_{v_i}); \vec{d}_{\texttt{Arg1}}(m_{v_i}); \vec{d}_{\texttt{loc}}(m_{v_i}); \vec{d}_{\texttt{time}}(m_{v_i})]$$

Symmetrically, we compute the argument vectors of an entity mention according to the events in which the entity mention plays a role.

This representation allows our model to directly compute the similarity between two mentions while considering a rich distributed representation of the current coreference clusters of their related arguments or predicates. Lee et al. (2012), on the other hand, modeled the dependencies between event and entity clusters using only simple discrete features, indicating the number of coreferring arguments across clusters.

The final mention vector is a concatenation of the various features: $\vec{v}(m) = [\vec{c}(m); \vec{s}(m); \vec{d}(m)]$,

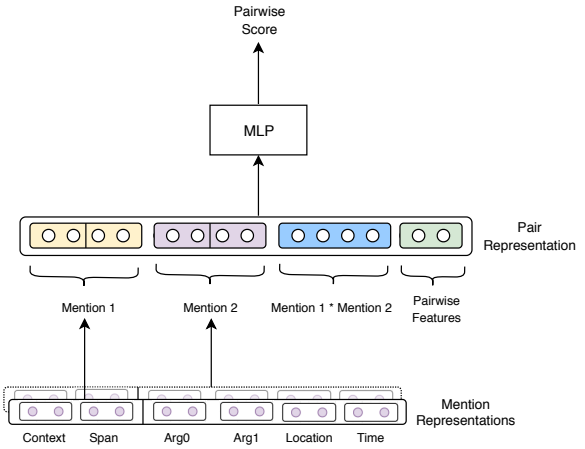

Figure 1: An illustration of the pairwise mention scorer. The bottom vectors are mention representations which include lexical and contextual features, and features derived from the mention's dependency on other mentions. The input to the network is a concatenation of two mention vectors with their element-wise multiplication and additional pairwise features.

as illustrated in Figure 1 (bottom row).

3.2 Mention-Pair Coreference Scorer

Figure 1 illustrates our pairwise mention scoring function $S(m_i, m_j)$ that returns a score denoting the likelihood that two mentions m_i and m_j are coreferring. We learn a separate function for entities (S_E) and for events (S_V), both trained identically as feed-forward neural networks. For the sake of simplicity, we describe them here as a single function $S(\cdot, \cdot)$.

The input to $S(m_i, m_j)$ is $\vec{v}_{i,j} = [\vec{v}(m_i); \vec{v}(m_j); \vec{v}(m_i) \circ \vec{v}(m_j); f(i,j)]$, where $\circ$ denotes an element-wise multiplication. Following Lee et al. (2012), we enrich our mention-pair representation with four pairwise binary features $f(i,j)$, indicating whether the two mentions have coreferring arguments (or predicates) in a given role (`Arg0`, `Arg1`, `location`, and `time`). We encode each binary feature as 50-dimensional embedding to increase its signal.

To train S_E we take as training examples all pairs of entity mentions that belong to different entity clusters in the current predicted configuration E_t. The gold label for a given pair (m_i, m_j) is set to 1 if they belong to the same gold cluster, and to 0 otherwise. We train it using binary cross entropy as the loss function. S_V is trained symmetrically.

3.3 Inference

Figure 2 describes our model step-by-step: the left part is the training procedure, while the right part is the inference procedure. The differences between the two procedures are highlighted. We first focus on the inference procedure (right), which gets as input the document set D, the pairwise mention scorers S_E and S_V, and the gold standard mentions.[2]

The algorithm operates over each topic separately. To that end, we start by applying document clustering using the K-Means algorithm, yielding a set of topics T. For a given topic t, the algorithm uses the gold entity and event mentions to build initial clusters. Event clusters V_t are initialized to singletons (line 2). Similarly to Lee et al. (2012), entity clusters E_t are initialized to the output of a within-document entity coreference resolution system (line 3).[3] Our iterative algorithm alternates between entity and event clustering, incrementally constructing the final clustering configuration (lines 4-12).

When the algorithm focuses on entities, it starts with updating the entity representations according to the event clusters in the current configuration, V_t (line 6). This update includes the recreation of argument vectors for each entity mention, as described in Section 3.1. We use agglomerative clustering that greedily merges multiple cluster pairs with the highest cluster-pair scores (line 8) until the scores are below a pre-defined threshold δ_2. The algorithm starts with high-precision merges, leaving less precise decisions to a latter stage, when more information becomes available. We define the cluster-pair score as the average mention linkage score: $S_{cp}(c_i, c_j) = \frac{1}{|c_i| \cdot |c_j|} \cdot \sum_{m_i \in c_i} \sum_{m_j \in c_j} S(m_i, m_j)$. The same steps are repeated for events (lines 10-12), and repeat iteratively until no merges are available or up to a predefined number of iterations (line 4).

3.4 Training

The training steps are similarly described in the left part of Figure 2. At each iteration, we train two updated scorer functions S_E (line 7) and S_V (line 11). Since our representation requires a clustering configuration, we use a training procedure that simulates the inference step. The training examples for each scorer change between iterations

[2] We follow the setup of Kenyon-Dean et al. (2018) and use the gold standard mentions (see Section 4).

[3] This reduces the search space, and decouples cross-document entity resolution from the within-document variant. The latter consists of phenomena such as pronoun resolution that are already handled well by existing tools.

Algorithm 1 Train

Require: D: document set
M^e, M^v: gold entity/event mentions
T: gold topics (document clusters)
$\{E_t\}_{t \in T}$: gold within-doc entity clusters
$G(\cdot)$: gold mention to cluster assignment

for $t \in T$ **do**
  $V_t \leftarrow$ SingletonEvents(t, M^v)

  while $\exists$ meaningful cluster-pair merge **do**
    // Entities
    $E_t \leftarrow$ UpdateJointFeatures(V_t)
    $S_E \leftarrow$ TrainMentionPairScorer(E_t, G)
    $E_t \leftarrow$ MergeClusters(S_E, E_t)
    // Events
    $V_t \leftarrow$ UpdateJointFeatures(E_t)
    $S_V \leftarrow$ TrainMentionPairScorer(V_t, G)
    $V_t \leftarrow$ MergeClusters(S_V, V_t)
return S_E, S_V

Algorithm 2 Inference

Require: D: document set
M^e, M^v: gold entity/event mentions
$S_E(\cdot, \cdot)$: pairwise entity mention scorer
$S_V(\cdot, \cdot)$: pairwise event mention scorer

$T \leftarrow$ ClusterDocuments(D)
for $t \in T$ **do**
  $V_t \leftarrow$ SingletonEvents(t, M^v)
  $E_t \leftarrow$ PredWithinDocEntityCoref(t, M^e)
  while $\exists$ meaningful cluster-pair merge **do**
    // Entities
    $E_t \leftarrow$ UpdateJointFeatures(V_t)

    $E_t \leftarrow$ MergeClusters(S_E, E_t)
    // Events
    $V_t \leftarrow$ UpdateJointFeatures(E_t)

    $V_t \leftarrow$ MergeClusters(S_V, V_t)
return $\{E_t\}_{t \in T}, \{V_t\}_{t \in T}$

Figure 2: Overview of the training algorithm (left) and the inference algorithm (right). The differences between the two procedures are highlighted.

based on cluster-pair merges occurred in previous iterations. This allows our model to be trained on various predicted clustering configurations that are gradually improved during the training.

The training procedure differs from the inference procedure by using the gold standard topic clusters and by initializing the entity clusters with the gold standard within-document coreference clusters. We do so in order to reduce the noise during training.

3.5 Implementation Details

Our model is implemented in PyTorch (Paszke et al., 2017), using the ADAM optimizer (Kingma and Ba, 2014) with a minibatch size of 16. We initialize the word-level representations to the pre-trained 300 dimensional GloVe word embeddings (Pennington et al., 2014), and keep them fixed during training. The character representations are learned using an LSTM with hidden size 50. We initialized them with pre-trained character embeddings[4]. Each scorer consists of a sigmoid output layer and two hidden layers with 4261 neurons activated by ReLU function (Nair and Hinton, 2010).

[4]Available at https://github.com/minimaxir/char-embeddings

We set the merging threshold in the training step to $\delta_1 = 0.5$. We tune the threshold for inference step on the validation set to $\delta_2 = 0.5$. To cluster documents into topics at inference time, we use the K-Means algorithm implemented in Scikit-Learn (Pedregosa et al., 2011). Documents are represented using TF-IDF scores of unigrams, bigrams, and trigrams, excluding stop words. We set $K = 20$ based on the Silhouette Coefficient method (Rousseeuw, 1987), which successfully reconstructs the number of test sub-topics. During inference, we use Stanford CoreNLP (Manning et al., 2014) to initialize within-document entity coreference clusters.

Identifying Predicate-Argument Structures. To extract relations between events and entities we follow previous work (Lee et al., 2012; Yang et al., 2015; Choubey and Huang, 2017) and extract predicate-argument structures using SwiRL (Surdeanu et al., 2007), a semantic role labeling (SRL) system. To increase the coverage we apply additional heuristics:

- Since SwiRL only identifies verbal predicates, we follow Lee et al. (2012) and consider nominal event mentions with possesors ("*Amazon's*

	Train	Validation	Test	Total
# Topics	25	8	10	43
# Sub-topics	50	16	20	86
# Documents	574	196	206	976
# Sentences	1037	346	457	1840
# Event mentions	3808	1245	1780	6833
# Entity mentions	4758	1476	2055	8289
# Event chains	1527	409	805	2741
# Entity chains	1286	330	608	2224

Table 1: ECB+ statistics (including singleton clusters). The split to topics is as follows - Train: 1, 3, 4, 6-11, 13-17, 19-20, 22, 24-33; validation: 2, 5, 12, 18, 21, 23, 34, 35; test: 36-45.

acquisition") as predicates and their `Arg0`.

- We use the spaCy dependency parser (Honnibal and Montani, 2017) to identify verbal event mentions whose subject and object are entities, and add those entities as their `Arg0` and `Arg1` roles, respectively.
- Following Lee et al. (2012), for a given event mention, we consider its closest left (right) entity mention as its `Arg0` (`Arg1`) role.

4 Experimental Setup

We use the ECB+ corpus, which is the largest dataset consisting of within- and cross-document coreference annotations for entities and events.

We follow the setup of Cybulska and Vossen (2015b), which was also employed by Kenyon-Dean et al. (2018). This setup uses a subset of the annotations which has been validated for correctness by Cybulska and Vossen (2014) and allocates a larger portion of the dataset for training (see Table 1). Since the ECB+ corpus only annotates a part of the mentions, the setup uses the gold-standard event and entity mentions rather, and does not require specific treatment for unannotated mentions during evaluation.

A different setup was carried out by Yang et al. (2015) and Choubey and Huang (2017). They used the full ECB+ corpus, including parts with known annotation errors. At test time, they rely on the output of a mention extraction tool (Yang et al., 2015). To address the partial annotation of the corpus, they only evaluated their systems on the subset of predicted mentions which were also gold mentions. Finally, their evaluation setup was criticized by Upadhyay et al. (2016) for ignoring singletons (cluster with a single mention), effectively making the task simpler; and for evaluating each sub-topic separately, which entails ignoring incorrect coreference links across sub-topics.

Evaluation Metrics. We use the official CoNLL scorer (Pradhan et al., 2014),[5] and report the performance on the common coreference resolution metrics: **MUC** (Vilain et al., 1995), $\mathbf{B^3}$ (Bagga and Baldwin, 1998a), **CEAF-e** (Luo, 2005), and **CoNLL** $\mathbf{F_1}$, the average of the 3 metrics.

5 Baselines

We compare our full model to published results on ECB+, available for event coreference only, as well as to a disjoint variant of our model and a deterministic lemma baseline.[6]

CLUSTER+LEMMA. We first cluster the documents to topics (Section 3.3), and then group mentions within the same document cluster which share the same head lemma. This baseline differs from the lemma baseline of Kenyon-Dean et al. (2018) which is applied across topics.

CV (Cybulska and Vossen, 2015a) is a supervised method for event coreference, based on discrete features. They first cluster documents to topics, and then cluster coreferring mentions within each topic cluster. Events are represented using information about participants, time and location, while documents are represented as "bag-of-events". We compare to their best reported results, differing from the CV baseline in Kenyon-Dean et al. (2018) which refers to the partial model that uses the same annotations in terms of subcomponents of the event structure.

KCP (Kenyon-Dean et al., 2018) is a neural network-based model for event coreference. They encode an event mention and its context into a vector and use it to cluster mentions. The model does not cluster documents to topics as a pre-processing step, but instead encodes the document as part of the mention representation, aiming to avoid spurious cross-topic coreference links thanks to distant document representations.

CLUSTER+KCP To tease apart the contribution of our document clustering component from that of the rest of the model, we add a variant of the KCP model which relies on our document clustering component as a pre-processing step. During inference, we restrict their model to clustering

[5] http://conll.github.io/reference-coreference-scorers/

[6] We do not compare our work to Yang et al. (2015) and Choubey and Huang (2017), since they used another incomparable evaluation setup, as discussed in Section 4.

Model	MUC R	MUC P	MUC F_1	B^3 R	B^3 P	B^3 F_1	CEAF-e R	CEAF-e P	CEAF-e F_1	CoNLL F_1
CLUSTER+LEMMA	71.3	83	76.7	53.4	84.9	65.6	70.1	52.5	60	67.4
DISJOINT	76.7	80.8	78.7	63.2	78.2	69.9	65.3	58.3	61.6	70
JOINT	78.6	80.9	79.7	65.5	76.4	70.5	65.4	61.3	63.3	**71.2**

Table 2: Combined within- and cross-document entity coreference results on the ECB+ test set.

Model	MUC R	MUC P	MUC F_1	B^3 R	B^3 P	B^3 F_1	CEAF-e R	CEAF-e P	CEAF-e F_1	CoNLL F_1
Baselines										
CLUSTER+LEMMA	76.5	79.9	78.1	71.7	85	77.8	75.5	71.7	73.6	76.5
CV (Cybulska and Vossen, 2015a)	71	75	73	71	78	74	-	-	64	73
KCP (Kenyon-Dean et al., 2018)	67	71	69	71	67	69	71	67	69	69
CLUSTER+KCP	68.4	79.3	73.4	67.2	87.2	75.9	77.4	66.4	71.5	73.6
Model Variants										
DISJOINT	75.5	83.6	79.4	75.4	86	80.4	80.3	71.9	75.9	78.5
JOINT	77.6	84.5	80.9	76.1	85.1	80.3	81	73.8	77.3	**79.5**

Table 3: Combined within- and cross-document event coreference results on the ECB+ test set.

	CoNLL F_1	Δ
Joint model	79.5	
– Pairwise binary features	79.4	-0.1
– Dependent mentions vectors	78.6	-0.9
– Both	78.5	-1.0

Table 4: Ablations of the joint modeling parts in our architecture. CoNLL F_1 score is reported for combined within- and cross-document event coreference.

mentions only within the same document cluster. Accordingly, we re-trained their model using the gold document clusters for hyper-parameters tuning to fit this cluster-based setting.

DISJOINT. A variant of our model which uses only the span and context vectors to build mention pair representations, ablating joint features.

We do not compare our work directly to Lee et al. (2012) since it was evaluated on a different corpus and using a different evaluation setup. Instead, we compare to CV and KCP, more recent models which reported their results on the ECB+ dataset.

With respect to entity coreference, to the best of our knowledge, our work is the first to publish entity coreference results on the ECB+ dataset. We therefore only compare our performance to that of the lemma baseline and our disjoint model.

6 Results

Table 2 presents the performance of our method with respect to entity coreference. Our joint model improves upon the strong lemma baseline by 3.8 points in CoNLL F_1 score.

Table 3 presents the results on event coreference. Our joint model outperforms all the baselines with a gap of 10.5 CoNLL F_1 points from the last published results (KCP), while surpassing our strong lemma baseline by 3 points.

The results reconfirm that the lemma baseline, when combined with effective topic clustering, is a strong baseline for CD event coreference resolution on the ECB+ corpus (Upadhyay et al., 2016). In fact, thanks to our near-perfect topic clustering on the ECB+ test set (Homogeneity: 0.985, Completeness: 0.982, V-measure: 0.984, Adjusted Rand-Index: 0.965), the CLUSTER+LEMMA baseline surpasses prior results on ECB+.

The results of CLUSTER+KCP again indicate that pre-clustering of documents to topics is beneficial, improving upon the KCP performance by 4.6 points, though still performing substantially worse than our joint model.

To test the contribution of joint modeling, we compare our joint model to its disjoint variant. We observe that the joint model performs better on both event and entity coreference. The performance gap is modest but significant with bootstrapping and permutation tests ($p < 0.001$).

We further ablate additional components from the full representation (Table 4). We show that each of our representation components contributes to performance, but the continuous vector components representing semantic dependency to other mentions are stronger than the pairwise binary features originally used by Lee et al. (2012).

7 Analysis

7.1 Error Analysis

To analyze the errors made by our joint model we sampled 50 event mentions and 50 entity mentions

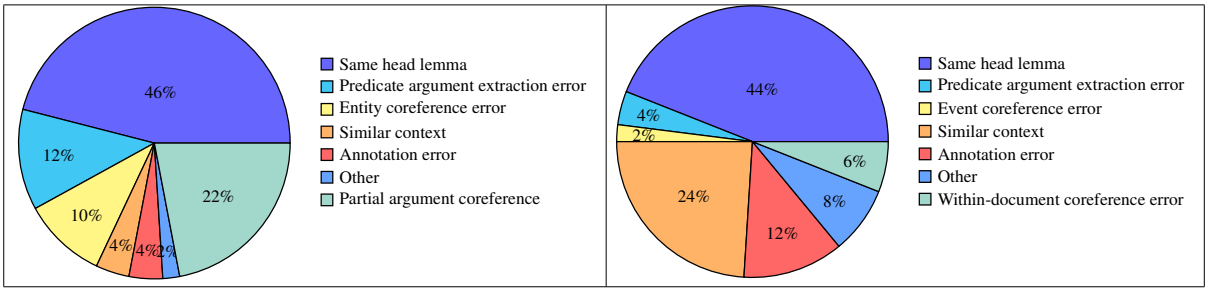

Figure 3: Event coreference errors (left) and entity coreference errors (right).

that were clustered incorrectly, i.e. where their predicted cluster contained at least 70% of mentions that are not in their gold cluster.

Figure 3 shows a pie chart for each mention type, manually categorized to error types, suggesting future areas for improvement. For both entities and events, mentions were often clustered incorrectly with other mentions that share the same head lemma. Errors in the extraction of the predicate-argument structures accounted for 12% of the errors in events and 4% for entities, e.g. marking *dozens* as the `Arg0` of *devastated* in "*dozens in a region devastated by the quake*".

The joint features caused 10% of the event errors and 2% of the entity errors, where two non-coreferring event mentions were clustered to the same event cluster based on their entity arguments that were incorrectly predicted as coreferring, and vice versa. For example, the event *shakes* in "*earthquake shakes Lake County*" and "*earthquake shakes Northern California*" was affected by the wrong coreference clustering of "*Lake County*" and "*Northern California*".

We also found mentions that were wrongly clustered together based on contextual similarity (24% for entities, 4% for events) as well as some annotation errors (12% and 4%). The within-document entity coreference system caused additional 6% of entity errors. Finally, 22% of the event errors were caused by event mentions sharing coreferring arguments. This may happen for instance when similar events occur at different times ("*The earthquake struck at about **9:30** a.m. and had a depth of **2.7** miles, according to the **USGS**.*" vs. "*The earthquake struck at about **7:30** a.m. and had a depth of **1.4** miles, according to the **USGS**.*").

7.2 Mention Representation Components

To understand the contribution of each component in the mention representation to the clustering, we visualize them. We focus on events, and sample 7 gold clusters from the test set that have at least 5 mentions each. We then compute t-SNE projections (Maaten and Hinton, 2008) of the full mention representation, only the context vector, and only the semantically-dependent mentions vector (top, middle, and bottom parts of Figure 4). In all the 3 graphs, each point refers to an event mention and its color represents the mention's gold cluster. The full mention representations (top) yield visibly better clusters, but the context vectors (middle) are also quite accurate, emphasizing the importance of modeling context for resolving coreference. The semantically-dependent mentions vectors (bottom) are less accurate on their own, yet, they manage to separate well some clusters even without access to the mention span itself, and based only on the predicate-argument structures.

8 Conclusion

We presented a neural approach for resolving cross-document event and entity coreference. We represent a mention using its text, context, and—inspired by the joint model of Lee et al. (2012)—we make an event mention representation aware of coreference clusters of entity mentions to which it is related via predicate-argument structures, and vice versa. Our model achieves state-of-the-art results, outperforming previous models by 10.5 CoNLL F_1 points on events, and providing the first cross-document entity coreference results on ECB+. Future directions include investigating ways to minimize the pipeline errors from the extraction of predicate-argument structures, and incorporating a mention prediction component, rather than relying on gold mentions.

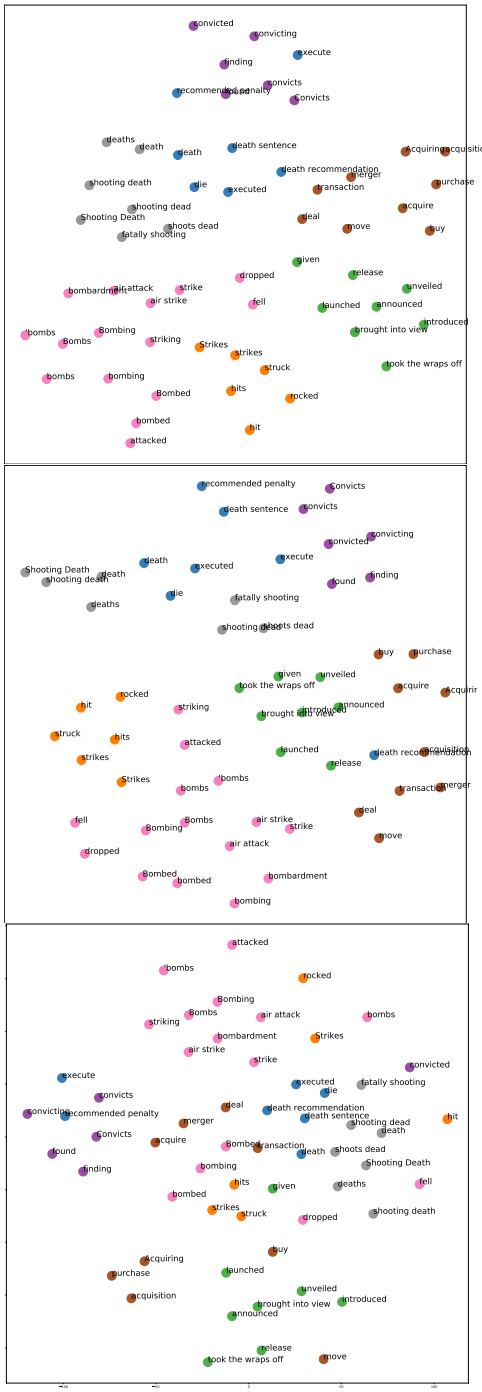

Figure 4: t-SNE projection of the full mention representation (top), context vector (middle) and dependent mention vector (bottom). Each point is an event mention, colored according to its gold cluster.

Acknowledgments

We would like to thank Jackie Chi Kit Cheung for the insightful comments. This work was supported in part by an Intel ICRI-CI grant, the Israel Science Foundation grant 1951/17, the German Research Foundation through the German-Israeli Project Cooperation (DIP, grant DA 1600/1-1), and a grant from Reverso and Theo Hoffenberg.

References

Amit Bagga and Breck Baldwin. 1998a. 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.

Amit Bagga and Breck Baldwin. 1998b. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98/COLING '98, pages 79–85, Stroudsburg, PA, USA. Association for Computational Linguistics.

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.

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

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.

Kevin Clark and Christopher D. Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany. Association for Computational Linguistics.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *Proceedings of the 9th international conference on Language Resources and Evaluation (LREC)*.

Agata Cybulska and Piek Vossen. 2015a. " bag of events" approach to event coreference resolution. supervised classification of event templates. *Int. J. Comput. Linguistics Appl.*, 6(2):11–27.

Agata Cybulska and Piek Vossen. 2015b. Translating granularity of event slots into features for event coreference resolution. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 1–10, Denver, Colorado. Association for Computational Linguistics.

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, Seattle, Washington, USA. Association for Computational Linguistics.

Sourav Dutta and Gerhard Weikum. 2015. Cross-document co-reference resolution using sample-based clustering with knowledge enrichment. *Transactions of the Association for Computational Linguistics*, 3:15–28.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

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.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. 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 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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, Jeju Island, Korea. 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.

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.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

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.

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

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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.

Marten Postma, Filip Ilievski, and Piek Vossen. 2018. Semeval-2018 task 5: Counting events and participants in the long tail. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 70–80, New Orleans, Louisiana. Association for Computational Linguistics.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–35, Baltimore, Maryland. Association for Computational Linguistics.

Delip Rao, Paul McNamee, and Mark Dredze. 2010. Streaming cross document entity coreference resolution. In *Coling 2010: Posters*, pages 1050–1058, Beijing, China. Coling 2010 Organizing Committee.

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Mihai Surdeanu, Lluís Màrquez, Xavier Carreras, and Pere R Comas. 2007. Combination strategies for semantic role labeling. *Journal of Artificial Intelligence Research*, 29:105–151.

Shyam Upadhyay, Nitish Gupta, Christos Christodoulopoulos, and Dan Roth. 2016. Revisiting the evaluation for cross document event coreference. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1949–1958, Osaka, Japan. The COLING 2016 Organizing Committee.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding*, pages 45–52. Association for Computational Linguistics.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association of Computational Linguistics*, 6:287–302.

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.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.