Joint Entity and Event Coreference Resolution across Documents

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, Dan Jurafsky
Stanford University, Stanford, CA 94305
{heeyoung,recasens,angelx,mihais,jurafsky}@stanford.edu

Abstract

We introduce a novel coreference resolution system that models entities and events jointly. Our iterative method cautiously constructs clusters of entity and event mentions using linear regression to model cluster merge operations. As clusters are built, information flows between entity and event clusters through features that model semantic role dependencies. Our system handles nominal and verbal events as well as entities, and our joint formulation allows information from event coreference to help entity coreference, and vice versa. In a cross-document domain with comparable documents, joint coreference resolution performs significantly better (over 3 CoNLL F1 points) than two strong baselines that resolve entities and events separately.

1 Introduction

Most coreference resolution systems focus on entities and tacitly assume a correspondence between entities and noun phrases (NPs). Focusing on NPs is a way to restrict the challenging problem of coreference resolution, but misses coreference relations like the one between *hanged* and *his suicide* in (1), and between *placed* and *put* in (2).

1. (a) **One of the key suspected Mafia bosses arrested yesterday** has hanged himself.
   (b) Police said **Lo Presti** had hanged himself.
   (c) His suicide appeared to be related to clan feuds.
2. (a) **The New Orleans Saints** placed **Reggie Bush** on the injured list on Wednesday.
   (b) **Saints** put **Bush** on I.R.

As (1c) shows, NPs can also refer to events, and so corefer with phrases other than NPs (Webber, 1988). By being anchored in spatio-temporal dimensions, events represent the most frequent referent of verbal elements. In addition to time and location, events are characterized by their participants or arguments, which often correspond with discourse entities. This two-way feedback between events and their arguments (or entities) is the core of our approach. Since arguments play a key role in describing an event, knowing that two arguments corefer is useful for finding coreference relations between events, and knowing that two events corefer is useful for finding coreference relations between entities. In (1), the coreference relation between *One of the key suspected Mafia bosses arrested yesterday* and *Lo Presti* can be found by knowing that their predicates (i.e., *has hanged* and *had hanged*) corefer. On the other hand, the coreference relations between the arguments *Saints* and *Bush* in (2) helps to determine the coreference relation between their predicates *placed* and *put*.

In this paper, we take a holistic approach to coreference. We annotate a corpus with cross-document coreference relations for nominal and verbal mentions. We focus on both intra and inter-document coreference because this scenario is at the same time more challenging and more relevant to real-world applications such as news aggregation. We use this corpus to train a model that jointly addresses references to both entities and events across documents. The contributions of this work are the following:

- We introduce a novel approach for entity and event coreference resolution. At the core of

our approach is an iterative algorithm that cautiously constructs clusters of entity and event mentions using linear regression to model cluster merge operations. Importantly, our model allows information to flow between clusters of both types through features that model context using semantic role dependencies.

- We annotate and release a new corpus with coreference relations between both entities and events across documents. The relations annotated are both intra and inter-document, which more accurately models real-world scenarios.

- We evaluate our cross-document coreference resolution system on this corpus and show that our joint approach significantly outperforms two strong baselines that resolve entities and events separately.

2 Related Work

Entity coreference resolution is a well studied problem with many successful techniques for identifying mention clusters (Ponzetto and Strube, 2006; Haghighi and Klein, 2009; Stoyanov et al., 2009; Haghighi and Klein, 2010; Raghunathan et al., 2010; Rahman and Ng, 2011, inter alia). Most of these techniques focus on matching compatible noun pairs using various syntactic and semantic features, with efforts targeted toward improving features and clustering models.

Prior work showed that models that jointly resolve mentions across multiple entities result in better performance than simply resolving mentions in a pairwise fashion (Denis and Baldridge, 2007; Poon and Domingos, 2008; Wick et al., 2008; Lee et al., 2011, inter alia). A natural extension is to perform coreference jointly across both entities and events. Yet there has been little attempt in this direction.

We know of only limited work that incorporates event-related information in entity coreference, typically by incorporating the verbs in context as features. For instance, Haghighi and Klein (2010) include the governor of the head of nominal mentions as features in their model. Rahman and Ng (2011) also used event-related information by looking at which semantic role the entity mentions can have and the verb pairs of their predicates. We confirm that such features are useful but also show that the complementary features for verbal mentions lead to even better performance, especially when event and entity clusters are jointly modeled.

Compared to the extensive work on entity coreference, the related problem of event coreference remains relatively under-explored, with minimal work on how entity and event coreference can be considered jointly on an open domain. Early work on event coreference for MUC (Humphreys et al., 1997; Bagga and Baldwin, 1999) focused on scenario-specific events. More recently, there have been approaches that looked at event coreference for wider domains. Chen and Ji (2009) proposed using spectral graph clustering to cluster events. Bejan and Harabagiu (2010) proposed a nonparametric Bayesian model for open-domain event resolution. However, most of this prior work focused only on event coreference, whereas we address both entities and events with a single model. Humphreys et al. (1997) considered entities as well as events, but due to the lack of a corpus annotated with event coreference, their approach was only evaluated implicitly in the MUC-6 template filling task. To our knowledge, the only previous work that considered entity and event coreference resolution jointly is He (2007), but limited to the medical domain and focused on just five semantic categories.

3 Architecture

Following the intuition introduced in Section 1, our approach iteratively builds clusters of event and entity mentions jointly. As more information becomes available (e.g., finding out that two verbal mentions have arguments that belong to the same entity cluster), the features of both entity and event mentions are re-generated, which prompts future clustering operations. Our model follows a cautious (or "baby steps") approach, which we previously showed to be successful for entity coreference resolution (Raghunathan et al., 2010; Lee et al., 2011). However, unlike our previous work, which used deterministic rules, in this paper we learn a coreference resolution model using linear regression. Algorithm 1 summarizes the flow of the proposed algorithm. We detail its steps next. We describe the training procedure in Section 4 and the features used in Section 5.

Algorithm 1: Joint Coreference Resolution

input : set of documents $\mathcal{D}$
input : coreference model Θ
// clusters of mentions:
1 $\mathcal{E}$= {}
// clusters of documents:
2 $\mathcal{C}$ = clusterDocuments($\mathcal{D}$)
3 **foreach** document cluster c **in** $\mathcal{C}$ **do**
  // all mentions in one doc cluster:
4 $\mathcal{M}$ = extractMentions(c)
  // singleton mention clusters:
5 $\mathcal{E}'$ = buildSingletonClusters($\mathcal{M}$)
  // high-precision deterministic sieves:
6 $\mathcal{E}'$ = applyHighPrecisionSieves($\mathcal{E}'$)
  // iterative event/entity coreference:
7 **while** $\exists\, e_1, e_2 \in \mathcal{E}' s.t.\ score(e_1, e_2, \Theta) > 0.5$ **do**
8 $(e_1, e_2) = \arg\max_{e_1, e_2 \in \mathcal{E}'} \text{score}(e_1, e_2, \Theta)$
9 $\mathcal{E}' = \text{merge}(e_1, e_2, \mathcal{E}')$
  // pronoun sieve:
10 $\mathcal{E}'$ = applyPronounSieve($\mathcal{E}'$)
  // append to global output:
11 $\mathcal{E} = \mathcal{E} + \mathcal{E}'$
output : $\mathcal{E}$

3.1 Document Clustering

Our approach starts with several steps that reduce the search space for the actual coreference resolution task. The first is document clustering, which clusters the set of input documents ($\mathcal{D}$) into a set of document clusters ($\mathcal{C}$). In the subsequent steps we only cluster mentions that appear in the same document cluster. We found this to be very useful in practice because, in addition to reducing the search space, it provides a word sense disambiguation mechanism based on corpus-wide topics. For example, without document clustering, our algorithm may decide to cluster two mentions of the verb *hit*, but knowing that one belongs to a cluster containing earthquake reports and the other to a cluster with reports on criminal activities, this decision can be avoided.[1]

Any non-parametric clustering algorithm can be used in this step. In this paper, we used the algorithm proposed by Surdeanu et al. (2005). This algorithm is an Expectation Maximization (EM) variant where the initial points (and the number of clusters) are selected from the clusters generated by a hierarchical agglomerative clustering algorithm using geometric heuristics. This algorithm performs well on our data. For example, in the training dataset, only two topics (handling different earthquake events) are incorrectly merged into the same cluster.

[1]Since different mentions of the verb *say* in the same topic might refer to different events, they are only merged if they have coreferent arguments.

3.2 Mention Extraction

In this step (4 in Algorithm 1) we extract nominal, pronominal, and verbal mentions. We extract nominal and pronominal mentions using the mention identification component in the publicly downloadable Stanford coreference resolution system (Raghunathan et al., 2010; Lee et al., 2011). We consider as verbal mentions all words whose part of speech starts with `VB`, with the exception of some auxiliary/copulative verbs (*have*, *be* and *seem*). For each of the identified mentions we build a singleton cluster (step 5 in Algorithm 1).

Crucially, we do not make a formal distinction between entity and event mentions. This distinction is not trivial to implement (e.g., is the noun *earthquake* an entity or an event mention?) and an imperfect classification would negatively affect the following coreference resolution. Instead, we simply classify mentions into verbal or nominal, and use this distinction later during feature generation (Section 5). To compare event nouns (e.g., *development*) with verbal mentions, the "derivationally related form" relation in WordNet is used.

3.3 High-precision Entity Resolution Sieves

To further reduce the problem's search space, in step 6 of Algorithm 1 we apply a set of high-precision filters from the Stanford coreference resolution system. This system is a collection of deterministic models (or "sieves") for entity coreference resolution that incorporate lexical, syntactic, semantic, and discourse information. These sieves are applied from higher to lower precision. As clusters are built, information such as mention gender and number is propagated across mentions in the same cluster, which helps subsequent decisions. The Stanford system obtained the highest score at the CoNLL-2011 shared task on English coreference resolution.

For this step, we selected all the sieves from the Stanford system with the exception of the pronoun resolution sieve. All the remaining sieves (listed in Table 1) have high precision because they employ linguistic heuristics with little ambiguity, e.g.,

High-precision sieves
Discourse processing sieve
Exact string match sieve
Relaxed string match sieve
Precise constructs sieve (e.g., appositives)
Strict head match sieves
Proper head noun match sieve
Relaxed head matching sieve

Table 1: Deterministic sieves in step 6 of Algorithm 1.

one sieve clusters together two entity mentions only when they have the same head word. Note that all these heuristics were designed for within-document coreference. They work well in our context because we apply them in individual document clusters, where the one-sense-per-discourse principle still holds (Yarowsky, 1995).

Importantly, these sieves do not address verbal mentions. That is, all verbal mentions are still in singleton clusters after this step. Furthermore, none of these sieves use features that facilitate the joint resolution of nominal and verbal mentions (e.g., features from semantic role frames). All these limitations are addressed next.

3.4 Iterative Entity/Event Resolution

In this stage (steps 7 – 9 in Algorithm 1), we construct entity and event clusters using a cautious or "baby steps" approach. We use a single linear regressor (Θ) to model cluster merge operations between both verbal and nominal clusters. Intuitively, the linear regressor models the quality of the merge operation, i.e., a score larger than 0.5 indicates that more than half of the mention pairs introduced by this merge are correct. We discuss the training procedure that yields this scoring function in Section 4. In each iteration, we perform the merge operation that has the highest score. Once two clusters are merged (step 9) we regenerate all the mention features to reflect the current clusters. We stop when no merging operation with an overall benefit is found.

This iterative procedure is the core of our joint coreference resolution approach. This algorithm transparently merges both entity and event mentions and, importantly, allows information to flow between clusters of both types as merge operations take place. For example, assume that during iteration i we merge the two *hanged* verbs in the first example in Section 1 (because they have the same lemma). Because of this merge, in iteration $i+1$ the nominal mentions *Lo Presti* and *One of the key suspected Mafia bosses* have the same semantic role for verbs assigned to the same cluster. This is a strong hint that these two nominal mentions belong to the same cluster. Indeed, the feature that models this structure received one of the highest weights in our linear regression model (see Section 7).

3.5 Pronoun Sieve

Our approach concludes with the pronominal coreference resolution sieve from the Stanford system. This sieve is necessary because our current resolution algorithm ignores mention ordering and distance (i.e., in step 7 we compare all clusters regardless of where their mentions appear in the text). As previous work has proved, the structure of the text is crucial for pronominal coreference (Hobbs, 1978). For this reason, we handle pronouns outside of the main algorithm block.

4 Training the Cluster Merging Model

Two observations drove our choice of model and training algorithm. First, modeling the merge operation as a classification task is not ideal, because only a few of the resulting clusters are entirely correct or incorrect. In practice, most of the clusters will contain some mention pairs that are correct and some that are not. Second, generating training data for the merging model is not trivial: a brute force approach that looks at all the possible combinations is exponential in the number of mentions. This is both impractical and unnecessary, as some of these combinations are unlikely to be seen in practice.

We address these observations with Algorithm 2. The algorithm uses gold coreference labels to train a linear regressor that models the quality of the clusters produced by merge operations. We define the quality score q of a new cluster as the percentage of new mention pairs (i.e., not present in either one of the clusters to be merged) that are correct:

$$q = \frac{links_{correct}}{links_{correct} + links_{incorrect}} \quad (1)$$

where $links_{(in)correct}$ is the number of newly introduced (in)correct pairwise mention links when two clusters are merged.

Algorithm 2: Training Procedure

input : set of documents $\mathcal{D}$
input : correct mention clusters $\mathcal{G}$
1 $\mathcal{C}$ = clusterDocuments($\mathcal{D}$)
// linear regression coreference model:
2 Θ = assignInitialWeights($\mathcal{C}, \mathcal{G}$)
// repeat for T epochs:
3 **for** *t = 1* **to** *T* **do**
// training data for linear regressor:
4 $\Gamma = \{\}$
5 **foreach** document cluster c **in** $\mathcal{C}$ **do**
6 $\mathcal{M}$ = extractMentions(c)
7 $\mathcal{E}$ = buildSingletonClusters($\mathcal{M}$)
8 $\mathcal{E}$ = applyHighPrecisionSieves($\mathcal{E}$)
// gather training examples
// as clusters are built:
9 **while** $\exists\, e_1, e_2 \in \mathcal{E} s.t.\ sco(e_1, e_2, \Theta) > 0.5$ **do**
10 **forall the** $e'_1, e'_2 \in \mathcal{E}$ **do**
11 q = qualityOfMerge($e'_1, e'_2, \mathcal{G}$)
12 Γ = append(e'_1, e'_2, q, Γ)
13 $(e_1, e_2) = \arg\max_{e_1, e_2 \in \mathcal{E}} \text{sco}(e_1, e_2, \Theta)$
14 $\mathcal{E}$ = merge($e_1, e_2, \mathcal{E}$)
// train using data from last epoch:
15 Θ' = trainLinearRegressor(Γ)
// interpolate with older model:
16 $\Theta = \lambda\Theta + (1 - \lambda)\Theta'$
output : Θ

We address the potential explosion in training data size by considering only merge operations that are likely to be inspected by the algorithm as it runs. To achieve this, Algorithm 2 repeatedly runs the actual clustering algorithm (as given by the current model Θ) over the training dataset (steps 5 – 14).[2] When the algorithm iteratively constructs its clusters (steps 9 – 14), we generate training data from all possible cluster pairs available during a particular iteration (steps 10 – 12). For each pair, we compute its score using Equation 1 (step 11) and add it to the training corpus Γ (step 12). Note that this avoids inspecting many of the possible cluster combinations: once a cluster is built (e.g., during the previous iterations or by the deterministic sieves in step 8), we do not generate training data from its members, but rather treat it as an atomic unit. On the other hand, our approach generates more training data than online learning, which trains using only the actual decisions taken during inference in each iteration (i.e., the pair (e_1, e_2) in step 13).

After each epoch we have a new training corpus Γ, which we use to train the new linear regression model Θ' (step 15), which is then interpolated with the old one (step 16).

Our training procedure is similar in spirit to transformation based learning (TBL) (Brill, 1995). Similarly to TBL, our approach repeatedly applies the model over the training data and attempts to minimize the error rate of the current model. However, while TBL learns rules that directly minimize the current error rate, our approach achieves this indirectly, by incorporating the reduction in error rate in the score of the generated datums. This allows us to fit a linear regression to this task, which, as discussed before, is a better model for this task.

Just like any hill-climbing algorithm, our approach has the risk of converging to a local maximum. To mitigate this risk, we do not initialize our model Θ with random weights, but rather use hints from the deterministic sieves. This procedure (listed in step 2) runs the high-precision sieves introduced in Section 3.3 and, just like the data generation loop in Algorithm 2, creates training examples from the clusters available after every merge operation. Since these deterministic models address only nominal clusters, at the end we generate training data for events by inspecting all the pairs of singleton verbal clusters. Using this data, we train the initial linear regression model.

We trained our model using L2 regularized linear regression with a regularization coefficient of 1.0. We did not tune the regularization coefficient. We ran the training algorithm for 10 epochs, although we observed minimal changes after three epochs. We tuned the interpolation weight (λ) to a value of 0.7 using our development corpus.

5 Features

We list in Table 2 the features used by the linear regression model. As the table indicates, our feature set relies heavily on semantic roles, which were extracted using the SwiRL semantic role labeling (SRL) system (Surdeanu et al., 2007).[3] Because SwiRL addresses only verbal predicates, we extended it to handle nominal predicates. In this

[2] We skip the pronoun sieve here because it does not affect the decisions taken during the iterative resolution steps.

[3] http://www.surdeanu.name/mihai/swirl/

Feature Name	Applies to Entities (E) or Events (V)	Description and Example
Entity Heads	E	Cosine similarity of the head-word vectors of two clusters. The head-word vector stores the head words of all mentions in a cluster and their frequencies. For example, the vector for the three-mention cluster {*Barack Obama*, *President Obama*, *US president*}, is {*Obama*:2, *president*:1}.
Event Lemmas	V	Cosine similarity of the lemma vectors of two clusters. For example, the lemma vector for the cluster {*murdered*, *murders*, *hitting*} is {*murder*:2, *hit*:1}.
Links between Synonyms	E, V	The percentage of newly-introduced mention links after the merge that are WordNet synonyms (Fellbaum, 1998). For example, when merging the following two clusters, {*hit*, *strike*} and {*strike*, *join*, *say*}, two out of the six new links are between words that belong to the same WordNet synset: (*hit – strike*) and (*strike – strike*).
Number of Coreferent Arguments or Predicates	E, V	The total number of shared arguments and predicates between mentions in the two clusters. We use the cluster IDs of the corresponding arguments/predicates to check for identity. For example, when comparing the event clusters {*bought*} and {*acquired*}, extracted from the sentences [*AMD*]$_{\texttt{Arg0}}$ *bought* [*ATI*]$_{\texttt{Arg1}}$ and [*AMD*]$_{\texttt{Arg0}}$ *acquired* [*ATI*]$_{\texttt{Arg1}}$, the value of this feature is 2 because the two mentions share one `Arg0` and one `Arg1` argument (assuming that the clusters {*AMD*, *AMD*} and {*ATI*, *ATI*} were previously created). For entity clusters, this feature counts the number of coreferent predicates. In addition to PropBank-style roles, for event mentions we also include the closest left and right entity mentions in order to capture any arguments missed by the SRL system.
Coreferent Arguments in a Specific Role?	E, V	Indicator feature set to 1 if the two clusters have at least one coreferent argument in a given role. We generate one variant of this feature for each argument label, e.g., `Arg0`, `Arg1`, etc. For example, the value of this feature for `Arg0` for the clusters {*bought*} and {*acquired*} in the above example is 1.
Coreferent Predicate in a Specific Role?	E	Indicator feature set to 1 if the two clusters have at least one coreferent predicate for a given role. For example, for the clusters {*the man*} and {*the person*}, extracted from the sentences *helped* [*the man*]$_{\texttt{Arg1}}$ and *helped* [*the person*]$_{\texttt{Arg1}}$, the value of this feature is 1 if the two *helped* verbs were previously clustered together.
2nd Order Similarity of Mention Words	E	Cosine similarity of vectors containing words that are distributionally similar to words in the cluster mentions. We built these vectors by extracting the top-ten most-similar words in Dekang Lin's similarity thesaurus (Lin, 1998) for all the nouns/adjectives/verbs in a cluster. For example, for the singleton cluster {*a new home*}, we construct this vector by expanding *new* and *home* to: {*new:1*, *original:1*, *old:1*, *existing:1*, *current:1*, *unique:1*, *modern:1*, *different:1*, *special:1*, *major:1*, *small:1*, *home:1*, *house:1*, *apartment:1*, *building:1*, *hotel:1*, *residence:1*, *office:1*, *mansion:1*, *school:1*, *restaurant:1*, *hospital:1* }.
Number; Animacy; Gender; NE Label	E	Cosine similarity of number, gender, animacy, and NE label vectors. For example, the number and gender vectors for the two-mention cluster {*systems*, *a pen*} are Number = {singular:1, plural:1}, Gender = {neutral:2}.

Table 2: List of features used when comparing two clusters. If any of the two clusters contains a verbal mention we consider the merge an operation between event (V) clusters; otherwise it is a merge between entity (E) clusters. We append to all entity features the suffix `Proper` or `Common` based on the type of the head word of the first mention in each of the two clusters. We use the suffix `Proper` only if both head words are proper nouns.

paper we used a single heuristic: the possessor of a nominal event's predicate is marked as its `Arg0`, e.g., *Logan* is the `Arg0` to *run* in *Logan's run*.[4]

[4] A principled solution to this problem is to use an SRL system for nominal predicates trained using NomBank (Meyers et al., 2004). We will address this in future work.

We extracted named entity labels using the named entity recognizer from the Stanford CoreNLP suite.

6 Evaluation

6.1 Corpus

The training and test data sets were derived from the EventCorefBank (ECB) corpus[5] created by Bejan and Harabagiu (2010) to study event coreference since standard corpora such as OntoNotes (Pradhan et al., 2007) contain a small number of annotated event clusters. The ECB corpus consists of 482 documents from Google News clustered into 43 topics, where a topic is described as a seminal event. The reason for including comparable documents was to increase the number of cross-document coreference relations. Bejan and Harabagiu (2010) only annotated a selection of events.

For the purpose of our study, we extended the original corpus in two directions: (i) fully annotated sentences, and (ii) entity coreference relations. In addition, we removed relations other than coreference (e.g., subevent, purpose, related, etc.) that had been originally annotated. We revised and completed the original annotation by annotating every entity and event in the sentences that were (partially) annotated. The annotation was performed by four experts, using the Callisto annotation tool.[6] The annotation guidelines and the generated corpus are available here.[7]

Our annotation of the ECB corpus followed the OntoNotes (Pradhan et al., 2007) standard for coreference annotation, with a few extensions to handle events. For nouns, we annotated full NPs (with all modifiers), excluding appositive phrases and nominal predicates. Only premodifiers that were proper nouns or possessive phrases were annotated. For events, we annotated the semantic head of the verb phrase. We extended the OntoNotes guidelines by also annotating singletons (but we do not score them; see below), and by including all events mentions (not only those mentioned at least once with an NP). This required us to be specific with respect to:

[5] http://faculty.washington.edu/bejan/data/ECB1.0.tar.gz

[6] http://callisto.mitre.org

[7] http://nlp.stanford.edu/pubs/jcoref-corpus.zip

	Training	Dev	Test	Total
# Topics	12	3	28	43
# Documents	112	39	331	482
# Entities	459	46	563	1068
# Entity Mentions	1723	259	3465	5447
# Events	300	30	444	774
# Event Mentions	751	140	1642	2533

Table 3: Corpus statistics.

⟨ENTITY COREFID="26"⟩ A publicist ⟨/ENTITY⟩ ⟨EVENT COREFID="4"⟩ says ⟨/EVENT⟩ ⟨ENTITY COREFID="23"⟩ Tara Reid ⟨/ENTITY⟩ has ⟨EVENT COREFID="3"⟩ checked ⟨/EVENT⟩ ⟨ENTITY COREFID="23"⟩ herself ⟨/ENTITY⟩ ⟨EVENT COREFID="3*"⟩ into ⟨/EVENT⟩ ⟨ENTITY COREFID="28"⟩ rehab ⟨/ENTITY⟩.

Figure 1: Annotation example.

Light verbs Verbs such as *give* and *make* followed by a noun (e.g., *make an offer*) were not annotated, but the noun was.

Phrasal verbs We annotated the verb together with the preposition or adverb (e.g., *check in*).

Idioms They were annotated with all their elements (e.g., *booze it up*).

The first topic was annotated by all four annotators as burn-in. Afterwards, annotation disagreements were resolved between all annotators and the next three topics were annotated again by all four annotators to measure agreement. Following Passonneau (2004), we computed an inter-annotator agreement of $\alpha = 0.55$ (Krippendorff, 2004) on these three topics, indicating moderate agreement among the annotators. Given the complexity of the task, we consider this to be a good score. For example, the average of the CoNLL F1 between any two annotators is 73.58, which is much higher than the system scores reported in the literature.

After annotating the four topics, disagreements were resolved again and all the documents in the four topics were corrected to match the consensus. The rest of the corpus was split between the four annotators, and each document was annotated by a single annotator. Figure 1 shows an example. Table 3 shows the corpus statistics, including the training, development (dev) and test set splits. The dev topics were used for tuning the interpolation parameter λ from Section 4.

System		MUC R	MUC P	MUC F1	B^3 R	B^3 P	B^3 F1	CEAF-ϕ_4 R	CEAF-ϕ_4 P	CEAF-ϕ_4 F1	BLANC R	BLANC P	BLANC F1	CoNLL F1
Baseline 1 Wo/ SRL	Entity	47.4	72.3	57.2	44.1	**82.7**	57.5	**42.5**	21.9	28.9	60.1	78.3	64.8	47.9
	Event	56.0	56.8	56.4	59.8	71.9	65.3	32.2	31.6	31.9	63.5	68.8	65.7	51.2
	Both	49.9	75.4	60.0	44.9	**83.9**	58.5	46.2	23.3	31.0	60.9	81.2	66.1	49.8
Baseline 2 With SRL	Entity	52.7	**73.0**	61.2	48.6	80.8	60.7	41.8	24.1	30.6	63.4	78.4	68.2	50.8
	Event	59.2	57.0	58.1	62.3	70.8	66.3	31.5	33.2	32.3	65.4	68.0	66.6	52.2
	Both	54.5	**76.4**	63.7	48.7	82.6	61.3	**46.3**	25.5	32.9	63.9	81.1	69.2	52.6
This paper	Entity	**60.7**	70.6	**65.2**	**55.5**	74.9	**63.7**	39.3	**29.5**	**33.7**	**66.9**	**79.6**	**71.5**	**54.2**
	Event	**62.7**	**62.8**	**62.7**	**62.5**	**73.9**	**67.7**	**34.0**	**33.9**	**33.9**	**67.6**	**78.5**	**71.7**	**54.8**
	Both	**61.2**	75.9	**67.8**	**53.9**	79.0	**64.1**	45.2	**30.0**	**35.8**	**67.1**	**82.2**	**72.3**	**55.9**

Table 4: Performance of the two baselines and our model. We report scores for entity clusters, event clusters and the complete task using five metrics.

6.2 Evaluation

We use five coreference evaluation metrics widely used in the literature:

MUC (Vilain et al., 1995) Link-based metric which measures how many predicted and gold clusters need to be merged to cover the gold and predicted clusters, respectively.

B^3 (Bagga and Baldwin, 1998) Mention-based metric which measures the proportion of overlap between predicted and gold clusters for a given mention.

CEAF (Luo, 2005) Entity-based metric that, unlike B^3, enforces a one-to-one alignment between gold and predicted clusters. We employ the entity-based version of CEAF.

BLANC (Recasens and Hovy, 2011) Metric based on the Rand index (Rand, 1971) that considers both coreference and non-coreference links to address the imbalance between singleton and coreferent mentions.

CoNLL F1 Average of MUC, B^3, and CEAF-ϕ_4. This was the official metric in the CoNLL-2011 shared task (Pradhan et al., 2011).

We followed the CoNLL-2011 evaluation methodology, that is, we removed all singleton clusters, and apposition/copular relations before scoring.

We evaluated the systems on three different settings: only on entity clusters, only on event clusters, and on the complete task, i.e., both entities and events. Note that the gold corpus separates clusters into entity and event clusters (see Table 3), but our system does not make this distinction at runtime. In order to compute the entity-only and event-only scores in Table 4, we implemented the following procedure: (a) when scoring entity clusters, we removed all mentions that were found to be coreferent with at least one gold event mention and not coreferent with any gold entity mentions; and (b) we performed the opposite action when scoring event clusters. This procedure is necessary because our mention identification component is not perfect, i.e., it generates mentions that do not exist in the gold annotation. Furthermore, this procedure is conservative with respect to the clustering errors of our system, e.g., all spurious mentions that our system includes in a cluster with a gold entity mention are considered for the entity score, regardless of their gold type (event or entity).

6.3 Results

Table 4 compares the performance of our system against two strong baselines that resolve entities and events separately. Baseline 1 uses a modified Stanford coreference resolution system after our document clustering and mention identification steps. Because the original Stanford system implements only entity coreference, we extended it with an extra sieve that implements lemma matching for events. This additional sieve merges two verbal clusters (i.e., clusters that contain at least one verbal mention) or a verbal and a nominal cluster when at least two lemmas of mention head words are the same between clusters, e.g., *helped* and *the help*.

The second baseline adds two more sieves to Baseline 1. Both these sieves model entity and event

contextual information using semantic roles. The first sieve merges two nominal clusters when two mentions in the respective clusters have the same head words and two mentions (possibly with different heads) modify with the same role label two predicates that have the same lemma. For example, this sieve merges the clusters {*Obama*, *the president*} (seen in the text [*Obama*]$_{\texttt{Arg0}}$ *attended* and [*the president*]$_{\texttt{Arg1}}$ *was elected*) and {*Obama*} (seen in the text [*Obama*]$_{\texttt{Arg1}}$ *was elected*), because they share a mention with the same head word (*Obama*) and two mentions modify with the same role (`Arg1`) predicates with the same lemma (*elect*). The second sieve implements the complementary action for event clusters. That is, it merges two verbal clusters when at least two mentions have the same lemma and at least two mentions have semantic arguments with the same role label and the same lemma.

7 Discussion

The first block in Table 4 indicates that lemma matching is a strong baseline for event resolution. Most of the event scores for Baseline 1 are actually higher than the corresponding entity scores, which were obtained using the highest ranked system at the CoNLL-2011 shared task (Lee et al., 2011). Adding contextual information using semantic roles (Baseline 2) helps both entities and events. The CoNLL F1 for Baseline 2 increases almost 3 points for entities and 1 point for events. This demonstrates that local syntactico-semantic context is important for coreference resolution even in a cross-document setting and that the current state-of-the-art in SRL can model this context accurately.

The best scores (almost unanimously) are obtained by the model proposed in this paper, which scores 3.4 CoNLL F1 points higher than Baseline 2 for entities, and 2.6 points higher for events. For the complete task, our approach scores 3.3 CoNLL F1 points higher than Baseline 2, and 6.1 points higher than Baseline 1. This demonstrates that a holistic approach to coreference resolution improves the resolution of both entities and events more than models that address aspects of the task separately. To further understand our experiments, we listed the top five entity/event features with the highest weights in our model in Table 5. The table indicates that six out of the ten features serve the purpose of passing information between entity and event clusters. For example, the "Coreferent Argument for `Arg1`" feature is triggered when two event clusters have `Arg1` arguments that already belong to the same entity cluster. This allows information from previous entity coreference operations to impact future merges of event clusters. This is the crux of our iterative approach to joint coreference resolution.

Entity Feature	Weight
Entity Heads – Proper	1.10
Coreferent Predicate for `ArgM-LOC` – Common	0.45
Entity Heads – Common	0.36
Coreferent Predicate for `Arg0` – Proper	0.29
Coreferent Predicate for `Arg2` – Common	0.28

Event Feature	Weight
Event Lemmas	0.45
Coreferent Argument for `Arg1`	0.19
Links between Synonym	0.16
Coreferent Argument for `Arg2`	0.13
Number of Coreferent Arguments	0.07

Table 5: Top five features with the highest weights.

Finally, we performed an error analysis by manually evaluating 100 errors. We distinguished nine major types of errors. Their ratios together with a description and an example are given in Table 6.

This work demonstrates that an approach that jointly models entities and events is better for cross-document coreference resolution. However, our model can be improved. For example, document clustering and coreference resolution can be solved jointly, which we expect would improve both tasks. Furthermore, our iterative coreference resolution procedure (Algorithm 1) could be modified to account for mention ordering and distance, which would allow us to include pronominal resolution in our joint model, rather than addressing it with a separate deterministic sieve.

8 Conclusion

We have presented a holistic model for cross-document coreference resolution that jointly solves references to events and entities by handling both nominal and verbal mentions. Our joint resolution algorithm allows event coreference to help improve entity coreference, and vice versa. In addition, our iterative procedure, based on a linear regressor that models the quality of cluster merges, allows each

Error Type (Ratio)	Description Example
Pronoun resolution (36%)	The pronoun is incorrectly resolved by the pronominal sieve of the Stanford deterministic entity system. These errors include (only a small number of) event pronouns. He said *Timmons* aimed and missed **his** target.
Semantics beyond role frames (20%)	The semantics of the coreference relation cannot be captured by role frames or WordNet. Israeli forces on Tuesday killed *at least 40 people* ... The Israeli army said the UN school in the Jabaliya refugee camp was hit ... and that **the dead** included a number of Hamas militants.
Arguments of nominal events (17%)	The arguments of two nominal events are not detected and thus not coreferred. The attack on *the school* has caused widespread shock across Israel ... while Israeli forces on Tuesday killed at least 40 people during an attack on **a United Nations-run school in Gaza**.
Cascaded errors (7%)	Entities or events are not coreferred due to errors in a previous merge iteration in the same semantic frame. In the example below, we failed to link the two *die* verbs, which leads to the listed entity error. An Australian climber who survived two nights stuck on Mount Cook after seeing *his brother* die ... **Dr Mark Vinar, 43**, is presumed dead ...
Initial high-precision sieves (6%)	An error made by the initial high-precision entity resolution sieves is propagated to our model. Timmons told police he fired when he thought he saw someone in the other group reach for a gun ... 15-year-old Timmons was at the scene of the shooting and had **a gun**.
Phrasal verbs (6%)	The meaning of a phrasal verb is not captured. A relative unknown will *take over* the title role of Doctor Who ... But **the casting of Smith** is a stroke of genius.
Linear regression (4%)	Recall error made by the regression model when the features are otherwise correct. The Interior Department on Thursday *issued* "revised" regulations ... Interior Secretary Dirk Kempthorne **announced** major changes ...
Mention detection (3%)	The mention detection module detects a spurious mention. Police have arrested a man ... in the parking lot crosswalk at Sam's Club in Bloomington.
SRL (1%)	The SRL system fails to label the semantic role. In this example, *jail* is detected as the ArgM-MNR of *hanged* instead of ArgM-LOC. A Mafia boss in Palermo hanged himself in **jail**.

Table 6: Error analysis. Mentions to be resolved are in **bold face**, correct antecedents are in *italics*, and our system's predictions are underlined.

merging state to benefit from the previous merged entity and event mentions. This approach allows us to start with a set of high-precision coreference relations and gradually add new ones to increase recall.

The experimental evaluation shows that our coreference algorithm gives markedly better F1 for both entities and events, outperforming two strong baselines that handle entities and events separately, measured by all the standard measures: MUC, B^3, CEAF-ϕ_4, BLANC and the official CoNLL-2011 metric. This is noteworthy since each measure has been shown to place primary emphasis in evaluating a different aspect of the coreference resolution task.

Our system is tailored for cross-document coreference resolution on a corpus that contains news articles that repeatedly report on a smaller number of topics. This makes it particularly suitable for real-world applications such as multi-document summarization and cross-document information extraction. We also release our labeled corpus to facilitate extensions and comparisons to our work.

Acknowledgements

We acknowledge the support of Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the DARPA, AFRL, or the US government. MR is supported by a Beatriu de Pinós postdoctoral scholarship (2010 BP-A 00149) from Generalitat de Catalunya. AC is supported by a SAP Stanford Graduate Fellowship. We also gratefully thank Cosmin Bejan for sharing his code and the useful discussions.

References

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

Amit Bagga and Breck Baldwin. 1999. Cross-document event coreference: Annotations, experiments, and observations. In *Proceedings of the ACL 1999 Workshop on Coreference and Its Applications*, pages 1–8.

Cosmin Bejan and Sanda Harabagiu. 2010. Unsupervised Event Coreference Resolution with Rich Linguistic Features. In *Proceedings of ACL 2010*, pages 1412–1422.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: a case study in part of speech tagging. *Computational Linguistics*, 21(4):543–565.

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

Pascal Denis and Jason Baldridge. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. In *Proceedings of NAACL-HLT 2007*.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge.

Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of EMNLP 2009*, pages 1152–1161.

Aria Haghighi and Dan Klein. 2010. Coreference resolution in a modular, entity-centered model. In *Proceedings of HLT-NAACL 2010*, pages 385–393.

Tian He. 2007. *Coreference Resolution on Entities and Events for Hospital Discharge Summaries*. Thesis, Massachusetts Institute of Technology.

Jerry R. Hobbs. 1978. Resolving pronoun references. *Lingua*, 44(4):311–338.

Kevin Humphreys, Robert Gaizauskas, and Saliha Azzam. 1997. Event coreference for information extraction. In *Proceedings of the Workshop On Operational Factors In Practical Robust Anaphora Resolution For Unrestricted Texts*, pages 75–81.

Klaus Krippendorff. 2004. *Content Analysis: An Introduction to its Methodology*. Sage, Thousand Oaks, CA, second edition.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of CoNLL 2011: Shared Task*, pages 28–34.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL 1998*, pages 768–774.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of HLT-EMNLP 2005*, pages 25–32.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004. The NomBank project: an interim report. In *Proceedings of the HLT-NAACL 2004 Workshop on Frontiers in Corpus Annotation*, pages 24–31.

Rebecca Passonneau. 2004. Computing reliability for coreference annotation. In *Proceedings of LREC 2004*, pages 1503–1506.

Simone Paolo Ponzetto and Michael Strube. 2006. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proceedings of HLT-NAACL 2006*, pages 192–199.

Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with Markov logic. In *Proceedings of EMNLP 2008*, pages 650–659.

Sameer S. Pradhan, Lance Ramshaw, Ralph Weischedel, Jessica MacBride, and Linnea Micciulla. 2007. Unrestricted coreference: Identifying entities and events in OntoNotes. In *Proceedings of ICSC 2007*, pages 446–453.

Sameer Pradhan, Lance Ramshaw, Mitchell Marcus, Martha Palmer, Ralph Weischedel, and Nianwen Xue. 2011. CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes. In *Proceedings of CoNLL 2011: Shared Task*, pages 1–27.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Chris Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of EMNLP 2010*, pages 492–501.

Altaf Rahman and Vincent Ng. 2011. Coreference resolution with world knowledge. In *Proceedings of ACL 2011*, pages 814–824.

William M. Rand. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

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

Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2009. Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. In *Proceedings of ACL-IJCNLP 2009*, pages 656–664.

Mihai Surdeanu, Jordi Turmo, and Alicia Ageno. 2005. A hybrid unsupervised approach for document clustering. In *Proceedings of KDD 2005*, pages 685–690.

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.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of MUC-6*, pages 45–52.

Bonnie Lynn Webber. 1988. Discourse deixis: reference to discourse segments. In *Proceedings of ACL 1988*, pages 113–122.

Michael L. Wick, Khashayar Rohanimanesh, Karl Schultz, and Andrew McCallum. 2008. A unified approach for schema matching, coreference and canonicalization. In *Proceedings of KDD 2008*, pages 722–730.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of ACL 1995*, pages 189–196.