Event Coreference Resolution with Multi-Pass Sieves

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

Abstract

Multi-pass sieve approaches have been successfully applied to entity coreference resolution and many other tasks in natural language processing (NLP), owing in part to the ease of designing high-precision rules for these tasks. However, the same is not true for event coreference resolution: typically lying towards the end of the standard information extraction pipeline, an event coreference resolver assumes as input the noisy outputs of its upstream components such as the trigger identification component and the entity coreference resolution component. The difficulty in designing high-precision rules makes it challenging to successfully apply a multi-pass sieve approach to event coreference resolution. In this paper, we investigate this challenge, proposing the first multi-pass sieve approach to event coreference resolution. When evaluated on the version of the KBP 2015 corpus available to the participants of EN Task 2 (Event Nugget Detection and Coreference), our approach achieves an Avg F-score of 40.32%, outperforming the best participating system by 0.67% in Avg F-score.

Keywords: event coreference resolution, discourse processing, information extraction

1. Introduction

Within-document event coreference resolution is the task of determining which event mentions in a text refer to the same real-world event. Compared to entity coreference resolution, event coreference resolution is not only much less studied, but it is arguably more challenging. The challenge stems in part from the fact that an event coreference resolver typically lies towards the end of the standard information extraction pipeline, assuming as input the noisy outputs of its upstream components. More specifically, an event coreference resolver assumes as input not only the event triggers, their types/subtypes, and their arguments, but also entity coreference information.

Different corpora have been used to train and evaluate event coreference resolvers, but as Liu et al. (2014) pointed out, not all of them were carefully annotated. As will be discussed in more detail in Section 2, OntoNotes and ECB have only be partially annotated with event coreference links. Among the publicly-available corpus, the ACE 2005 corpus is arguably the one that is most complete with respect to the annotation of event coreference links. In fact, the majority of recent work on event coreference was evaluated on the ACE 2005 corpus.

As an event coreference corpus, ACE 2005 has a major weakness: it adopts a strict notion of event identity. Specifically, two event mentions were annotated as coreferent if and only if "they had the same agent(s), patient(s), time, and location" (Song et al., 2015), and their event attributes (polarity, modality, genericity, and tense) are not incompatible. This is arguably an overly strict definition of event coreference, as some event mentions are intuitively coreferent even if their time and/or location arguments are not identical.

The new KBP 2015 event coreference corpus was created in response to the aforementioned weakness of the ACE 2005 corpus (Song et al., 2015). It was annotated using the Rich ERE guidelines, which are arguably more realistic in the sense that they mimic more closely a human's judgment of whether two event mentions are coreferent. There are at least three major differences between the ACE guidelines and the Rich ERE guidelines for annotating a document with event coreference chains. First, while ACE allows only single-word event triggers (main verbs, nouns, adjectives, adverbs), Rich ERE additionally allows multi-word phrases to be event triggers. For example, "laid off" is the event trigger in the sentence "Jane was laid off by XYZ Corp." As can be seen, just using "laid" as the trigger does not allow the event to be represented correctly. Second, while ACE allows at most one event mention to be triggered by a given word, Rich ERE allows the same word/phrase to trigger multiple event mentions with different types/subtypes. For instance, the word "murder" can trigger two event mentions, one with subtype Life.Die and the other with subtype Conflict.Attack. As can be seen, having only one of these two event mentions does not sufficiently represent the underlying events. Finally, and perhaps most importantly, while ACE adopts the aforementioned strict notion of event coreference, Rich ERE defines a relaxed coreference criterion, which allows two event mentions be coreferent as long as they *intuitively* refer to the same real-world event. For instance, the two event mentions "attack in Baghdad on Thursday" and "bombing in the Green Zone last week", though having different time and location expressions, are intuitively coreferent, and will be annotated as coreferent according to Rich ERE but not ACE. This relaxed notion of event coreference yields an event coreference task that is not only more realistic but also more challenging than that of ACE, since we can no longer rule out two event mentions as being coreferent simply on the grounds that their times and locations are different, for instance. Our goal is to work with this realistic version of the event coreference task and presenting one of the first results on the new KBP 2015 event coreference corpus. In particular, we propose a multi-pass sieve approach to

event coreference resolution. Multi-pass sieves were originally applied to entity coreference resolution (Raghunathan et al., 2010; Lee et al., 2013) and have then been successfully applied to many other tasks in natural language processing (NLP) such as temporal relation extraction (Chambers et al., 2014), spatial relation extraction (D'Souza and Ng, 2015b), and disorder mention normalization (D'Souza and Ng, 2015a). Though rarely explicitly mentioned, successful application of a sieve-based approach to a given task depends heavily on the extent to which high-precision rules can be designed for the task. For event coreference resolution, designing high-precision rules is by no means trivial. The reason is that, as mentioned above, an event coreference resolver typically assumes as input the noisy outputs of its upstream components. The difficulty in designing high-precision rules makes the successful application of a multi-pass sieve approach to event coreference resolution challenging.

In this paper, we address this challenge, proposing the first multi-pass sieve approach to event coreference resolution. When evaluated on the version of the KBP 2015 corpus available to the participants of the Event Nugget Detection and Coreference task, our approach achieves an Avg F-score of 40.32%, outperforming the best participating system by 0.67% in Avg F-score.

The rest of the paper is organized as follows. Section 2 presents an overview of related work on event coreference resolution. In Section 3, we describe our evaluation corpus, which is the corpus used in the official KBP 2015 Event Nugget Detection and Coreference task. Sections 4 and 5 discuss our baseline system and our multi-pass sieve approach to event coreference resolution. Finally, we present evaluation results in Section 6 and conclusions in Section 7.

2. Related Work

Early work on event coreference resolution was primarily evaluated on the MUC and ACE corpora, both of which contained within-document event coreference links. While event coreference research in MUC was limited to several scenarios such as terrorist attacks, management succession and resignation (e.g., Humphreys et al. (1997)), the ACE program takes a further step towards processing more fine-grained events. Most ACE event coreference resolvers are supervised, training a pairwise model to determine whether two event mentions are coreferent (e.g., Ahn (2006), Chen and Ng (2013; 2014)). Improvements to this standard approach include the use of (1) feature weighting to train a better model (McConky et al., 2012), and (2) graph-based clustering algorithms to produce event coreference clusters (e.g., Chen and Ji (2009), Sangeetha and Arock (2012)). Despite the successes of supervised approaches, Chen and Ng (2015) proposed an unsupervised probabilistic model for event coreference resolution that rivaled its supervised counterparts when evaluated on the ACE corpus.

There have also been attempts to evaluate within-document event coreference resolvers on other corpora, such as OntoNotes (Pradhan et al., 2007). For instance, Chen et al. (2011) trained multiple classifiers to handle coreference between event mentions of different syntactic types (e.g., verb-noun coreference, noun-noun coreference) on the OntoNotes corpus. However, since event coreference links and entity coreference links are not distinguished in OntoNotes, Chen et al. made the simplifying assumption that event coreference chains are all and only those coreference chains that involve at least one verb when performing event coreference on OntoNotes.

Researchers have also employed other corpora when evaluating their event coreference resolvers. For instance, Cybulska and Vossen (2012) performed event coreference on the Intelligence Community (IC) corpus using semantic relations (e.g., hyponymy relations extracted from WordNet). The IC corpus, which at the time of writing is not yet publicly available, is different from the MUC and ACE corpora in that it is annotated with not only *full* event coreference relations but also *partial* event coreference relations. Partial coreference is a term coined by Hovy et al. to refer to event relations that exhibit subtle deviation from the perfect identity of events (e.g., the subset relation, the membership relation). While all of the aforementioned work addresses the full event coreference task, a two-stage approach was recently proposed by Araki et al. (2014) to identify subevent relations from the IC corpus.

Bejan and Harabagiu (2010; 2014) evaluated their unsupervised nonparametric models on the EventCorefBank (ECB) corpus, which is composed of documents annotated with both within-document and cross-document event coreference links. Lee et al. (2012) extended the ECB corpus by annotating it with entity coreference links, which allow them to propose a "joint" method that iteratively performs entity coreference and event coreference by allowing one model to make use of the partial results produced so far for the other model in each iteration. While calling their approach a joint approach, they employ neither joint learning nor joint inference. A closer look at the ECB corpus reveals that within-document coreference links are only partially annotated (Liu et al., 2014): in almost all documents only the first few sentences are annotated with entity and event coreference links. In response to the missing links problem, the ECB+ corpus (Cybulska and Vossen, 2014b), an extension to ECB, was created. ECB+ was used by Yang et al. (2015) to evaluate their hierarchical distance-dependent Bayesian event coreference model.

More recently, Araki and Mitamura (2015) have evaluated their event coreference system on the ProcessBank corpus (Berant et al., 2014), a corpus of 200 paragraphs taken from a biology textbook. Specifically, they performed event trigger identification and event coreference resolution simultaneously using a structured perceptron.

The newest event coreference corpus is perhaps the one used in the KBP 2015 Event Nugget Detection and Coreference shared task. The teams that achieved the highest scores have adopted different strategies for this task. RPI's system viewed the event nugget coreference space as an undirected weighted graph in which the nodes represent all the event nuggets and the edge weight indicates coreference confidence between two event nuggets (Hong et al., 2015). LCC's system first determined the compatibility of each pair of event mentions in the document using a multi-stage pipeline and then employed a greedy iterative clustering algorithm to produce event hoppers (Monahan et al.,

Training Data	Newswire	Forum
Documents	81	77
Event mentions	2,219	4,319
Event hoppers	1,461	1874
Evaluation Data	**Newswire**	**Forum**
Documents	98	104
Event mentions	3,788	2,650
Event hoppers	2,440	1,685

Table 1: Statistics on the official KBP 2015 Event Nugget Detection and Coreference corpus.

2015). UI-CCG's system first modeled the similarity between two event mentions either by a supervised model or in an unsupervised fashion, and then made a coreference decision on each pair (Sammons et al., 2015).

3. Corpus and Task Definition

In this section, we introduce our corpus and the event coreference task.

Our evaluation corpus is the one used in the Event Nugget Detection and Coreference task in the TAC KBP 2015 Event Track (henceforth the KBP 2015 coreference corpus). This corpus is composed of two types of documents: newswire documents and discussion forum documents. Statistics on the corpus are shown in Table 1.

This corpus is annotated according to the Rich ERE annotation guidelines.[1] Rich ERE defines the following terminologies related to event detection and coreference:

- Event mention: an explicit occurrence of an event with or without participants. An Rich ERE event mention consists of a textual trigger, arguments or participants if exist and the event type/subtype.
- Event trigger: a string of text that most clearly expresses the occurrence of event, usually a word or a multi-word phrase
- Event argument: an entity or an argument filler that plays a certain role in an event.
- Event hopper: a group of event mentions that refer to the same event. They must have the same event type/subtype, but are allowed to have different arguments and triggers. It is a slightly relaxed standard of coreference compared to the ACE standard.

Despite the fact that the corpus is annotated with event arguments, the version of the corpus we employ in this paper (i.e., the version available to the participants of the Event Nugget Detection and Coreference Task in the KBP 2015 Event Track) is only annotated with event mentions, event triggers, and event hoppers. In particular, it does *not* contain any event argument annotations. Nevertheless, the KBP organizers have made available to the shared task participants a number of annotated corpora that the participants are allowed to use to train their systems. Some of these corpora are composed with newswire and discussion forum documents and are annotated with event mentions, triggers, arguments, and hoppers according to the Rich ERE guidelines. As we will see in the next section, we make use of two of these corpora for training our entity extractor and our argument identification and role determination classifier.

[1] See `http://cairo.lti.cs.cmu.edu/kbp/2015/event/annotation`.

French far-left killer$_{[Person]}$ {leaves}(EV1) jail$_{[Origin]}$ A former militant of the French far-left group Action Directe, Georges Cipriani$_{[Person]}$ {left}(EV2) prison$_{[Origin]}$ on parole on Wednesday$_{[Time]}$ after 23 years behind bars for two high-profile murders. A policeman at the scene confirmed to AFP the identity of Cipriani, 59, who$_{[Person]}$ {left}(EV3) the prison in Ensisheim in northeastern France$_{[Origin]}$wearing a leather jacket and with long white hair. Cipriani$_{[Person]}$ {left}(EV4) Ensisheim$_{[Origin]}$ in a police vehicle$_{[Instrument]}$ bound for an open prison near Strasbourg where the police officer said he was due to do community service including working at a food bank as part of his parole.

Table 2: Event coreference resolution example.

To better understand the aforementioned definitions, consider the text segment in Table 2. The trigger underlying each event mention is surrounded by curly brackets and marked with an identifier (e.g., the event mention "leaves" is marked with the identifier EV1), and its arguments are underlined with their roles in square brackets.[2] As we can see, this example contains four event mentions, all of which belong to the same event hopper because they have the same type (MOVEMENT) and subtype (TRANSPORT-PERSON) and intuitively refer to the same real-world event. Although some arguments are not identical, one can determine that these event mentions are coreferential when examining the surrounding contexts. For example, the *Origin* argument "Ensisheim" of EV4 and the *Origin* arguments "prison" and "jail" of EV1, EV2 and EV3 are different in granularity, and yet one can easily infer from the contexts *Cipriani left Ensisheim* and *Cipriani left the prison in Ensisheim* that they refer to the same event, for instance.

4. Baseline System

In this section, we describe our baseline system, which operates in three steps. First, it performs event mention detection, which involves detects all explicit mentioning of events with certain specified types in text (Section 4.1). Second, it performs event argument identification and role determination, which involves identifying the arguments of each event mention detected in the first step and assigning a semantic role to each participating argument (Section 4.2). Finally, it performs event coreference resolution on the event mentions extracted in the first step (Section 4.3), using the arguments detected in the second step as one of its knowledge sources.

[2] Recall that event argument annotations are not available in the KBP 2015 coreference corpus. They are shown in this example for ease of exposition only.

4.1. Event Trigger Identification and Subtyping

This component extracts event triggers and determines the type and subtype of each extracted trigger. In the KBP 2015 coreference corpus, there are nine event types and 38 event subtypes. A event trigger can be a single word or a multi-word phrase. We recast the task of identifying event triggers as a sequence labeling task, where we train CRFs using the CRF++ package on the training portion of the KBP 2015 coreference corpus.[3] As mentioned in the introduction, since each word can trigger multiple event mentions having different types/subtypes, we train one CRF for each type. Specifically, for classifier of type t_j, we create one instance for each word w_i, assigning it a class label that indicates whether it begins a trigger with subtype s_{jk} (B-s_{jk}), is inside a trigger with subtype s_{jk} (I-s_{jk}), begins a trigger with other types (B-$t_{m \neq j}$), is inside a trigger with other types (I-$t_{m \neq j}$) or is outside a trigger (O). Below we describe the 13 features used to represent w_i, which can be divided into three categories: lexical, syntactic and semantic.

Lexical: word unigrams (w_{i-2},w_{i-1},w_i,w_{i+1},w_{i+2}); word bigrams ($w_{i-1}w_i$, w_iw_{i+1}); word trigrams ($w_{i-2}w_{i-1}w_i$, $w_{i-1}w_iw_{i+1}$, $w_iw_{i+1}w_{i+2}$); the part-of-speech tag of w_i; lemmatized word unigrams, bigrams and trigrams.

Syntactic: depth of w_i's node in its syntactic parse tree; the path from the leaf node of w_i to the root in its syntactic parse tree; the phrase structure expanded by the parent of w_i's node; the phrase type of w_i's node. We compute the syntactic features based on the syntactic parse trees returned by Stanford's CoreNLP package (Manning et al., 2014).

Semantic: the WordNet synset id of w_i; the WordNet synset ids of w_i's hypernym, its parent, and its grandparent. When computing these semantic features, we only use the synset corresponding to w_i's first sense.

We improve the recall of event trigger detection in a postprocessing process as follows. First, we construct a wordlist containing triggers that appear infrequently (less than 10 times) in the training data and do not belong more than one subtype according to the training data. For example, the word "hijack" appears only a few times in the training data but is always labeled as "Conflict.Attack". Then, we extract any word as a trigger with the corresponding subtype as long as it appears in the wordlist.

4.2. Event Argument Identification and Role Classification

This component takes as inputs (1) a set of event mentions whose triggers were identified in the previous component (see Section 4.1) and (2) a set of candidate event arguments. For each event mention em, it identifies those candidate arguments that are the true arguments of em and assigns a role to each of its true arguments. In the rest of this subsection, we first describe how we extract the candidate event arguments for each event mention, and then show we identify and assign roles to its true arguments.

[3] https://taku910.github.io/crfpp/

		Newswire	Forum
LDC2015E29	Documents	48	43
	Entity mentions	2,751	4,906
LDC2015E68	Documents	–	95
	Entity mentions	–	12,570

Table 3: Statistics on LDC2015E29 and LDC2015E68.

4.2.1. Extracting Candidate Event Arguments

Event arguments can be entity mentions or argument fillers. Argument fillers correspond to specific event subtypes, meaning that they will only appear if the corresponding subtype lends itself to such information. In addition, argument fillers such as Title and Age provide little useful information for event coreference. For this reason, we only extract entity mentions as candidate event arguments.

To extract entity mentions, we train a CRF (using CRF++) on two of the annotated corpora made available to us by the KBP 2015 shared task organizers, LDC2015E29 and LDC2015E68, both of which are annotated with Rich ERE entity mentions. Statistics on those corpora are shown in Table 3.

We train the CRF to jointly identify and determine the semantic type of each entity mention. Specifically, we create one instance for each word w_i, assigning it a class label that indicates whether it begins an entity mention of type t_j (B-t_j) , is inside an entity of type t_j (I-t_j) or is outside an entity (O). In Rich ERE annotation, each entity is labeled with one of five semantic types: PER, ORG, GPE, LOC, and FAC, so under the IOB labeling scheme, there are 11 labels in total. Each token w_i is represented using the nine features, as described below:

Lexical: word unigrams, bigrams, and trigrams formed from w_i in a window of five.

Grammatical: the part-of-speech tag of w_i; whether w_i is in a NP or not; whether w_i is part of a pronoun, whether the first letter of w_i is in uppercase.

Semantic: the WordNet synset id of w_i; the WordNet synset ids of the w_i's hypernym, its parent, and its grandparent.

4.2.2. Identifying True Arguments and their Roles

We jointly learn the tasks of (1) identifying the true arguments of an event mention and (2) assigning a role to each of its true arguments. We train this classifier on the documents in LDC2015E29 and LDC2015E68, both of which are annotated with event arguments. To create training instances, we pair each true event mention em (i.e., event mention consisting of a true trigger) with each of em's candidate event arguments, considering an entity mention extracted by our CRF-based entity-mention extractor a candidate argument of em if it appears in the same sentence as em. If the candidate argument is indeed a true argument of em, the class label of the training instance is the argument's role. Otherwise, its class label is *None*. There are 27 labels in total, including 26 roles defined in the Rich ERE annotation and NONE. Each instance is represented by 13 features, as described below:

Basic: trigger subtype; type of candidate argument; head

word of candidate argument; event subtype + head word; event subtype + entity type; POS of trigger word.

Neighboring words: left/right neighbor word of the candidate argument; left/right neighbor word of the candidate argument + the word's POS; left/right neighbor word of the trigger + the word's POS.

Syntactic: the phrase structure obtained by expanding the parent of the trigger in the constituent parse tree; the phrase type of the trigger; the path from the candidate argument to the trigger in the constituent parse tree; the dependency path from the candidate argument to the trigger.

To create test instances, we pair each *candidate* event mention (i.e., an event mention whose trigger was identified in Section 4.1) with each of its candidate event arguments. The test instances are represented using the same set of features as the training instances.

4.3. Event Coreference Resolution

This component identifies event coreference links by combining a mention-pair model (Soon et al., 2001), which is a binary classifier that determines whether two event mentions are co-referring or not, with a closest-first single-link clustering algorithm, which selects as the antecedent of an event mention e the closest preceding event mention that is classified as coreferent with e. We train the mention-pair model using the libSVM software package (Chang and Lin, 2001) as follows. We first divide the training documents of the KBP 2015 coreference corpus into two sets: a 128-document *training* set for model training, and a 30-document *development* set for jointly tuning the regularization parameter C and the γ parameter associated with the RBF kernel.[4] Then we retrain the model on all 158 training documents using the learned parameters.

We create positive training instances by pairing each anaphoric event mention em with its closest antecedent and (2) negative training instances by pairing em with each of its preceding event mentions that is not coreferent with em. Each instance is representing using 22 features. We use Stanford CoreNLP package to extract the linguistic information needed to compute these features, including the part-of-speech tags, syntactic parse trees, dependency parse trees and entity coreference chains. As can be seen below, the 22 features can be divided into three groups. For convenience, we use em_2 to refer to an event mention to be resolved and em_1 to refer to a candidate antecedent of em_2.

Group 1 (Event Type and Subtype features). The four features in this group encode: whether em_1 and em_2 agree w.r.t. event type; whether they agree w.r.t. event subtype; the concatenation of their event types; and the concatenation of their event subtypes.

Group 2 (Event Trigger features). The ten features in this group encode: whether em_1 and em_2 have the same trigger; whether they have the same lemmatized trigger; whether the triggers of em_1 and em_2 or the hypernyms of these triggers are in the same synset in WordNet; the concatenation of their triggers; the concatenation of part-of-speech tags of their triggers; whether their triggers agree in number if they are nouns; whether their triggers have the same modifiers if they are nouns; whether their triggers are in the same entity coreference chain if they are nouns; the sentence distance between the triggers of em_1 and em_2; whether the triggers of em_1 and em_2 appear in a training document as a coreferent event mention pair.

Group 3 (Event Argument features). The eight features in this group encode: whether em_1 and em_2 have arguments of the same role; whether the arguments have the same head word; whether they are in the same coreference chains; whether they have the same modifiers; the roles and number of the arguments that only appear in em_1; and the roles and number of the arguments that only appear in em_2

5. A Multi-Pass Sieve Approach

In this section, we describe our multi-pass sieve approach to event coreference resolution. The sieve approach has been successfully applied to entity coreference resolution. To our knowledge, ours represents the first attempt to apply the sieve approach to event coreference resolution.

5.1. Brief Introduction to Sieves

A sieve is composed of one or more heuristic rules. Each rule extracts a coreference relation between two event mentions. Sieves are ordered by their precision, with the most precise sieve appearing first. To resolve a set of event mentions in a document, the resolver makes multiple passes over them. In the i-th pass, we process the event mentions in a test text from left to right. Each event mention encountered, em_2, is compared in turn to each preceding event mention, em_1, from right to left. If any of the rules in the i-th sieve posits the two as coreferent, we will select em_1 as the antecedent of em_2. Once an antecedent has been selected for em_2, we will process the next mention in the text. In other words, we will not select more than one antecedent for each event mention. If none of em_2's preceding event mentions is posited as coreferent with it, then em_2 will remain unresolved in the i-th pass. The partial clustering of event mentions generated in the i-th pass is then passed to the i+1-th pass. In this way, later passes can exploit the information computed by previous passes, but the decision make earlier cannot be overridden later.

In our approach, later sieves exploit the decisions made by the earlier sieves as follows. When two event mentions are posited as coreferent by a sieve, any argument extracted for one mention will be shared by the other mention. It is this sharing of argument among coreferent event mentions that will be exploited by the later sieves.

5.2. Sieves for Event Coreference

Given a test document, our sieve approach first extracts (1) the event mentions using the CRF described in Section 4.1 and (2) their arguments using the SVM classifier described in Section 4.2, and then employs the following sieves for event coreference resolution. The sieves we designed for processing newswire documents are slightly different from those for processing discussion forum documents, as described below.

5.2.1. Sieves for Newswire Documents

We employ the following six sieves. Note that whenever a rule posits two event mentions as coreferent, we merge the

[4] We attempted values of 2^{-1}, 2, 2^3, 2^5, 2^7 for C and 2^{-7}, 2^{-5}, 2^{-3}, 2^{-1}, 2 for γ.

clusters containing the two mentions.

1. **Newswire Headline sieve:** the design of this sieve is motivated by the journalistic nature of newswire documents. The first sentence in the newswire documents always contains a detailed explanation of the headline. This sieve posits an event mention in the headline and an event mention the first sentence as coreferent if they have the same subtype and their triggers are in the same WordNet synset.
2. **Strict Event Coreference sieve:** the design of this sieve is motivated by the strict event coreference criterion. Two mentions are posited as coreferent if they satisfy all of the following conditions: (a) they have the same subtype; (b) their triggers have the same lemmatized form; (c) at least one of their arguments of the same role are in the same entity coreference chain or are lexically identical (if they are non-pronominal); and (d) their triggers are in the same entity coreference chain if they are nouns.
3. **Strict Trigger Match sieve:** this sieve posits two event mentions with noun triggers as coreferent if they have the same subtypes and their triggers have the same lemma and same modifiers.
4. **Semantically Similar Triggers sieve:** this sieve relaxes the conditions in the Strict Event Coreference Sieve by deleting conditions (b) and (d), but it requires the triggers of the two mentions or the hypernyms of the triggers to be in the same WordNet synset.
5. **Known Coreferent Pairs sieve:** this sieve posits two event mentions as coreferent if they have the same subtypes and the underlying triggers have appeared in the training data as a coreferent event mention pair.
6. **Machine Learning sieve:** this sieve exploits the information provided by the baseline system described in the previous section. Specifically, two event mentions are posited as coreferent if their coreference probability exceeds a certain threshold according to the baseline mention-pair model. The threshold is tuned on the 30-document development set described in Section 4.3.[5]

5.2.2. Sieves for Discussion Forum Documents

For discussion forum documents, we employ essentially the same sieves except that we replace the first sieve with a sieve that posits two event mentions as coreferent if their triggers and the sentences containing them are identical. This sieve is motivated by the nature of a discussion forum where an author usually quotes a preceding post to which she wants to respond.

6. Evaluation

In this section, we evaluate our multi-pass sieve approach to event coreference resolution.

6.1. Experiment Setup

Corpora. As mentioned before, we use LDC2015E29 and LDC2915E68 to train our entity mention extractor and our event argument identification and role classification model. In addition, we use the training portion of the KBP 2015 coreference corpus for training our trigger identification and subtyping model and our mention-pair-based event coreference classifier. 20% of this training data is used to tune the threshold in the Machine Learning sieve.

Evaluation metrics. To evaluate event coreference performance, we employ four commonly-used coreference scoring measures as implemented in version 1.7 of the official scorer provided by the KBP 2015 organizers, namely MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), CEAF_e (Luo, 2005) and BLANC (Recasens and Hovy, 2011).[6] Each of these evaluation measures reports results in terms of recall (R), precision (P), and F-score (F). We also report event mention detection performance in terms of recall, precision and F-score, considering a mention correctly detected if it has an exact match with a gold mention in terms of boundary, event type, and event subtype.

6.2. Results and Discussion

We evaluate the baseline and our sieve-based approach in an *end-to-end* setting using automatically detected event mentions. Our event mention detection system (see Section 4.1) achieves scores of 50.50% (R), 66.60% (P), and 57.45% (F). It underperforms only one participating system, which achieves an F-score of 58.41%, in the official KBP 2015 evaluation.

Table 4 shows the results of the baseline system (row 1) and our sieve-based event coreference resolver (rows 2 to 7). As we can see, the baseline achieves an Avg F-score (the unweighted average of the F-scores of the four scoring measures) of 37.82%. The subsequent rows show the results when the six sieves are added incrementally into the system. When all six sieves are employed, our approach achieves an Avg F-score of 40.32%, yielding a statistically significant improvement of 2.5% absolute F-score over the baseline (paired t-test, $p < 0.05$). These results provide suggestive evidence that our multi-pass sieve approach to event coreference, which is a hybrid rule-based and learning-based approach, is superior to a pure learning-based approach. Equally importantly, our approach outperforms the best participating system in the official KBP 2015 evaluation, which achieves an F-score of 39.65%.

6.3. Qualitative Error Analysis

Next, we present an analysis of the major sources of error made by our sieve-based approach.

A major source of recall error stems from our system's inability to cluster event mentions that have few common features. Consider the sentence "Somali pirates said Saturday they had received a record nine million dollar ransom in a helicopter air drop for the release of a South Korean supertanker, Samho Dream, with 24 crew. 'The boat was freed this morning agter the payment of nine million dollars to my colleagues,' one of the pirates told AFP by telephone." In this example, "ransom" and "payment" are triggers of two coreferent event mentions, but our system failed to detect this coreferent pair for at least two reasons. First, the semantic similarity of these two triggers provides little evidence that the corresponding event mentions are coreferent.

[5] We attempted values of 0.5, 0.6, 0.7, 0.8, and 0.9 for the threshold.

[6] The official scorer is available at `http://cairo.lti.cs.cmu.edu/kbp/2015/event/scoring`.

	MUC			B^3			$CEAF_e$			BLANC			Avg
	R	P	F1	R	P	F1	R	P	F1	R	P	F1	F1
Baseline	29.26	50.78	37.13	39.34	53.88	45.48	35.85	41.66	38.54	23.82	40.93	30.12	37.82
Sieve 1	0.73	73.91	1.45	30.66	66.17	41.91	43.47	36.90	39.92	13.68	58.22	17.13	25.10
+ Sieve 2	6.26	53.70	11.22	31.88	65.02	42.78	43.21	38.65	40.80	15.15	49.55	19.92	28.68
+ Sieve 3	9.07	56.30	15.63	32.46	64.61	43.21	43.32	39.63	41.39	15.93	50.10	21.29	30.38
+ Sieve 4	11.06	52.35	18.27	32.98	63.59	43.44	42.59	40.00	41.25	16.50	48.31	22.22	31.29
+ Sieve 5	40.51	48.00	43.93	42.75	48.33	45.37	33.07	46.56	38.67	29.67	39.26	33.11	40.27
+ Sieve 6	40.68	48.08	44.07	42.82	48.29	45.39	33.04	46.60	38.67	29.72	39.27	33.14	40.32

Table 4: Event coreference results on the official KBP 2015 evaluation data.

Second, in order to know that their arguments "a South Korean supertanker" and "the boat" are coreferent, we need a deeper linguistic analysis.

A major source of precision error stems from our system's tendency to cluster event mentions whose triggers have the same lemma. Despite the fact that we employ a Semantically Similar Triggers sieve, additional background knowledge is needed to resolve these difficult cases.

7. Conclusion

We have presented a multi-pass sieve approach to the under-studied task of event coreference resolution. When evaluated on the version of the KBP 2015 corpus available to the participants of the Event Nugget Detection and Coreference task, our approach achieves an Avg F-score of 40.32%, outperforming the best participating system by 0.67% in Avg F-score.

8. Acknowledgments

We thank the three anonymous reviewers for their detailed and insightful comments on an earlier draft of this paper. This work was supported in part by NSF Grant IIS-1219142. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of NSF.

9. Bibliographical References

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

Araki, J. and Mitamura, T. (2015). Joint event trigger identification and event coreference resolution with structured perceptron. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2074–2080.

Araki, J., Liu, Z., Hovy, E., and Mitamura, T. (2014). Detecting subevent structure for event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 4553–4558.

Bagga, A. and Baldwin, B. (1998). Algorithms for scoring coreference chains. In *Proceedings of the Linguistic Coreference Workshop at The First International Conference on Language Resources and Evaluation*, pages 563–566.

Bejan, C. and Harabagiu, S. (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.

Bejan, C. A. and Harabagiu, S. (2014). Unsupervised event coreference resolution. *Computational Linguistics*, 40(2):311–347.

Berant, J., Srikumar, V., Chen, P.-C., Huang, B., Manning, C. D., Vander Linden, A., Harding, B., and Clark, P. (2014). Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Langauge Processing*, pages 1499–1510.

Chambers, N., Cassidy, T., McDowell, B., and Bethard, S. (2014). Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Chang, C.-C. and Lin, C.-J., (2001). *LIBSVM: A library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chen, Z. and Ji, H. (2009). Graph-based event coreference resolution. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*, pages 54–57.

Chen, C. and Ng, V. (2013). Chinese event coreference resolution: Understanding the state of the art. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, pages 822–828.

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

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

Chen, B., Su, J., Pan, S. J., and Tan, C. L. (2011). A unified event coreference resolution by integrating multiple resolvers. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 102–110.

Cybulska, A. and Vossen, P. (2012). Using semantic relations to solve event coreference in text. In *Proceedings of the LREC Workshop on Semantic Relations-II Enhancing Resources and Applications*, pages 60–67.

D'Souza, J. and Ng, V. (2015a). Sieve-based entity linking for the biomedical domain. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 297–302.

D'Souza, J. and Ng, V. (2015b). Sieve-based spatial relation extraction with expanding parse trees. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 758–768.

Hong, Y., Lu, D., Yu, D., Pan, X., Wang, X., Chen, Y., Huang, L., and Ji, H. (2015). RPI BLENDER TAC-KBP2015 System Description. In *Proceedings of the 2015 Text Analysis Conference*.

Humphreys, K., Gaizauskas, R., and Azzam, S. (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.

Lee, H., Recasens, M., Chang, A., Surdeanu, M., and Jurafsky, D. (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.

Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., and Jurafsky, D. (2013). Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916.

Liu, Z., Araki, J., Hovy, E., and Mitamura, T. (2014). Supervised within-document event coreference using information propagation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 4539–4544.

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

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

McConky, K., Nagi, R., Sudit, M., and Hughes, W. (2012). Improving event co-reference by context extraction and dynamic feature weighting. In *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, pages 38–43.

Monahan, S., Mohler, M., Tomlinson, M., Book, A., Gorelkin, M., Crosby, K. and Brunson, M. (2015). Populating a knowledge base with information about events. In *Proceedings of the 2015 Text Analysis Conference*.

Pradhan, S., Ramshaw, L., Weischedel, R., MacBride, J., and Micciulla, L. (2007). Unrestricted coreference: Identifying entities and events in OntoNotes. In *Proceedings of the International Conference on Semantic Computing*, pages 446–453.

Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., and Manning, C. (2010). A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501.

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

Sammons, M., Peng, H., Song, Y., Upadhyay, S., Tsai, C., Reddy, P., Roy, S., and Roth, D. (2015). Illinois CCG TAC 2015 event nugget, entity discovery and linking, and slot filler validation systems. In *Proceedings of the 2015 Text Analysis Conference*.

Sangeetha, S. and Arock, M. (2012). Event coreference resolution using mincut based graph clustering. *International Journal of Computing and Information Sciences*, pages 253–260.

Song, Z., Bies, A., Strassel, S., Riese, T., Mott, J., Ellis, J., Wright, J., Kulick, S., Ryant, N., and Ma, X. (2015). From light to rich ERE: Annotation of entities, relations, and events. In *Proceedings of the 3rd Workshop on EVENTS at the NAACL-HLT*, pages 89–98.

Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Vilain, M., Burger, J., Aberdeen, J., Connolly, D., and Hirschman, L. (1995). A model-theoretic coreference scoring scheme. In Proceedings of the Sixth Message Understanding Conference, pages 45–52.

Yang, B., Cardie, C., and Frazier, P. (2015). A hierarchical distance-dependent bayesian model for event coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:517–528.