Event Coreference Resolution: A Survey of Two Decades of Research

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

Abstract

Recent years have seen a gradual shift of focus from entity-based tasks to event-based tasks in information extraction research. Being a core event-based task, event coreference resolution is less studied but arguably more challenging than entity coreference resolution. This paper provides an overview of the major milestones made in event coreference research since its inception two decades ago.

1 Introduction

Compared to entity coreference resolution, event coreference resolution is less studied but arguably more challenging. To see its difficulty, consider the following example on *within-document* event coreference resolution, whose goal is to determine which event mentions in a document refer to the same real-world event:

> Georges Cipriani $\{\text{left}\}_{ev1}$ a prison in Ensisheim in northern France on parole on Wednesday. He $\{\text{departed}\}_{ev2}$ the prison in a police vehicle bound for an open prison near Strasbourg.

In this example, there are two event mentions, *ev1* and *ev2*, which are triggered by the words *left* and *departed* respectively. These event mentions are coreferent because they both refer to the same event of Cipriani leaving the prison.

Intuitively, for two event mentions to be coreferent, not only should they have the same event subtype, but their arguments should be compatible. In our example, *ev1* and *ev2* have the same subtype, Movement.Transport-Person, thus satisfying the subtype agreement constraint. As far as argument compatibility is concerned, note that an event mention has zero or more *arguments* (the event's participants), each of which plays a role. For instance, *ev1* has three arguments: *Georges Cipriani* is the PERSON argument, *a prison* is the ORIGIN argument, and *Wednesday* is its TIME argument. *ev2* also has three arguments, *He*, *the prison*, and *a police vehicle*, serving as its PERSON, ORIGIN, and INSTRUMENT arguments respectively. Since the two event mentions have two overlapping roles (i.e., PERSON and ORIGIN) and their arguments are (entity-)coreferent w.r.t. each of these roles, they satisfy the argument compatibility constraint.

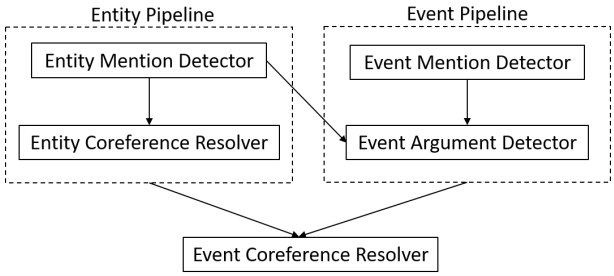

Figure 1: The standard information extraction pipeline

It should be easy to see from this example that to perform end-to-end event coreference resolution, one has to build an information extraction (IE) pipeline (cf. Figure 1) that involves (1) extracting the entity mentions from a given document (the *entity extraction* component) and determining which of them are coreferent (the *entity coreference* component); (2) extracting the event mentions by identifying their trigger words/phrases and determining which entity mentions are their arguments (the *event extraction* component); and (3) determining which event mentions are coreferent.

There are at least two reasons why event coreference resolution is potentially more challenging than entity coreference resolution. First, as we can see in Figure 1, while an entity coreference resolver has to assume as inputs the noisy outputs of an entity extraction component, an event coreference resolver has to assume as inputs the noisy outputs of a larger set of upstream components in both the entity pipeline and the event pipeline, each of which involves challenging tasks that are far from being solved. Second, while entity mentions are by and large noun phrases, event mentions are more diverse syntactic objects, including not only noun phrases (e.g., gerunds, event-denoting nouns) but also verb phrases, sentences and arguably whole paragraphs, which led the community to use both verbs and nouns as event mentions.

Despite its difficulty, event coreference resolution is the fundamental technology for consolidating the textual information about an event, which is crucial for essentially all high-level natural language processing (NLP) applications. For example, in IE, events and event coreference information have been used for template filling [Humphreys *et al.*, 1997] and automated population of knowledge bases [Ji and Grishman, 2011]. In topic detection and tracking, event corefer-

Corpora	Event type/subtypes	Event Arguments labeled?	Language	Document Type	WD/CD	Size (Approx. number of docs)
MUC	Typed	Yes	EN	news	WD	60 (MUC6), 50 (MUC7)
ACE	8 types, 33 subtypes	Yes	EN,CN	news, broadcast conversations, web blogs, and conversational telephone speech	WD	ACE 2005: 600 (EN), 500 (CN)
OntoNotes	Untyped	No	EN	Wall Street Journal (WSJ) news	WD,CD	600
ECB/ECB+	Untyped	Yes	EN	Google news	WD,CD	1000 (ECB+)
KBP	9 types, 38 subtypes	Yes	EN,CN,ES	news, discussion forum	WD	1000 (EN), 800 (CN), 400 (ES)

Table 1: Comparison of the publicly available event coreference-annotated corpora.

ence information is needed to identify new events in a stream of broadcast news stories [Allan *et al.*, 1998]. In event-based text summarization, event coreference information has been used to measure the similarity between two events, which in turn can be used to determine whether a sentence is salient or not [Li *et al.*, 2006]. Finally, event coreference information has also been used in other applications such as question answering [Narayanan and Harabagiu, 2004] and contradiction detection [De Marneffe *et al.*, 2008].

Our goal in this paper is to provide the AI audience with an overview of the major milestones made in event coreference research since its inception 20 years ago. Given the gradual shift of focus from entity-based tasks to event-based tasks in IE research in recent years, we believe that this timely survey will be of interest to AI researchers.

We conclude this section by mentioning that while the majority of work on event coreference resolution focuses on the within-document version of this task, there has also been work on *cross-document* event coreference, where the goal is to determine whether two event mentions in different documents refer to the same real-world event.

2 Corpora

In this section, we present five publicly-available corpora that have been widely used for training and evaluating event coreference resolvers. Table 1 compares these corpora along six dimensions, including (1) whether event (sub)types are annotated; (2) whether event arguments are annotated; (3) whether the documents are in English (EN), Chinese (CN), and/or Spanish (ES); (4) the types of documents; (5) whether the documents are annotated with within-document (WD) and/or cross-document (CD) event coreference chains; and (6) the approximate size in terms of the number of documents.

These corpora differ in another aspect that is not covered in the table: the definition of an *event*. As event coreference chains are annotated on top of event mentions, the definition of events (i.e., which event mentions to annotate) is relevant to event coreference because changing the definition could change the event coreference links in the resulting corpus. Below we discuss this and other aspects of the five corpora, presenting them in chronological order so that the reader can get a better idea of how the task has evolved over the years.

The Message Understanding Conferences (MUCs) produce the earliest corpora for supporting the event coreference task [MUC6, 1995; MUC7, 1998]. The MUC evaluations center around a "scenario", which is defined in terms of a key event type and various roles pertaining to it. Note that MUC does not define/evaluate event coreference officially: event coreference is (implicitly) a task that needs to be performed as part of the *scenario template* filling task. In this filling task, one has to fill *one* template (consisting of the various event roles/attributes) for each event mentioned in a document. Hence, given two event mentions in a document, one has to determine whether one or two templates should be filled by determining whether they are coreferent.

The ACE corpora are produced as part of the Automatic Content Extraction (ACE) evaluations. ACE 2005, the most widely used version of the ACE corpora for within-document event coreference evaluations, includes both English and Chinese documents. In ACE, an event is defined as "a specific occurrence of something that happens, often a change of state, involving participants" [LDC, 2005]. As in MUC, only events belonging to certain (sub)types are annotated in ACE, but ACE covers a larger set of (sub)types than MUC.

OntoNotes is a large-scale corpus that covers entities and events that are not limited to a predefined set of entity and event types [Pradhan *et al.*, 2007]. It provides both within- and cross-document entity and event coreference annotations. However, it does not specify which mentions are entity mentions and which are event mentions, nor does it annotate all event coreference chains: a chain is annotated if and only if at least one of its event mentions is nominal. It is partially for this reason that OntoNotes is less used for event coreference evaluation than the other corpora.

The EventCorefBank (ECB) corpus[1] [Bejan and Harabagiu, 2008] and its revised version, ECB 0.1[2] [Lee *et al.*, 2012], follow the TimeML specification, where events are characterized as "situations that happen or occur". Events can be expressed as punctual, durational, or stative predicates describing "states or circumstances in which something obtains or holds true" [Pustejovsky *et al.*, 2003]. Like OntoNotes, ECB contains both within- and cross-document event coreference links that belong to one of 43 event types. Since its focus is cross-document event coreference, within-document links are only partially annotated. The ECB+ corpus [Cybulska and Vossen, 2014] extends ECB 0.1 by incorporating more annotated documents and re-annotating the existing documents with the new annotation style. It also expands the definition by modeling events as a combination of four arguments, namely action, time, location, and participants.

As part of the TAC KBP event detection and coreference

[1] http://adi.bejan.ro/data/ECB1.0.tar.gz

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

evaluations, which started in 2015, several within-document multilingual event coreference-annotated corpora following the RichERE annotation style have been released [Song *et al.*, 2015]. The KBP corpora define a complex and hierarchical event structure that goes beyond any of the existing (and partially-overlapping) corpus. They follow the definition in the ACE corpora, but expand on taggability in several areas: a slightly expanded event ontology, the addition of generic and other (irrealis) event mentions, the addition of argumentless triggers for event mentions, additional attributes for contact and transaction events, double tagging of event mentions for multiple types/subtypes, and multiple tagging of event mentions for certain types of coordination.

3 Models

While early work on event coreference resolution has employed a rule-based approach [Humphreys *et al.*, 1997], virtually all recent work has adopted a learning-based approach, as described below.

3.1 Supervised Models

Mention-Pair Models

Following early entity coreference resolvers (e.g., Soon *et al.* [2001], Ng and Cardie [2002]), many event coreference resolvers adopt a two-step resolution framework. In the first step, a binary classifier (known as a *mention-pair model*) is used to determine whether two event mentions are coreferent. Mention-pair models are typically trained using an off-the-shelf learning algorithm, such as decision trees [Cybulska and Vossen, 2015], maximum entropy [Ahn, 2006; Chen and Ji, 2009], support vector machines [Chen and Ng, 2014], and deep neural networks [Nguyen *et al.*, 2016].

After training, the resulting mention-pair model can be applied to classify the test instances. However, these pairwise classification decisions could violate transitivity, which is an inherent property of the coreference relation. Hence, in the second step, a separate clustering mechanism is needed to coordinate the pairwise decisions and construct a partition. Some researchers employ *agglomerative clustering* algorithms, such as *closest-first* clustering (selecting as the antecedent of an event mention the closest preceding event mention that is classified as coreferent with it by the mention-pair model) and *best-first* clustering (selecting as the antecedent of an event mention the preceding coreferent event mention that has the highest coreference likelihood according to the mention-pair model) [Chen and Ng, 2014; Peng *et al.*, 2016]. Others employ *graph partitioning*. Specifically, given a test document, an undirected weighted graph is first constructed, where the nodes represent the event mentions in the document and the weight of an edge represents the coreference likelihood of the two nodes it connects. Then, a clustering algorithm, such as spectral clustering and divisive clustering, is used to obtain coreference clusters [Chen and Ji, 2009; Chen *et al.*, 2009; Sangeetha and Arock, 2012].

Improvements to this approach include using feature weighting to train a better model [McConky *et al.*, 2012] and training multiple classifiers to handle coreference between event mentions of different syntactic types [Chen *et al.*, 2011].

Generative Models

Though conceptually simple and extensively investigated, mention-pair models and the associated two-step approach suffer from *error propagation*, where errors made by a mention-pair model can propagate to the clustering step. To address this problem, Yang *et al.* [2015] propose a *supervised* nonparametric generative model for event coreference resolution, building on the framework of the distance-dependent Chinese restaurant process. The model has several key appealing properties. As a clustering model, event mentions are directly assigned to incrementally built coreference clusters. As a nonparametric model, the number of clusters does not need to be known a priori. As a Bayesian model, it can exploit priors, which in this case encode the knowledge provided by a mention-pair model. Finally, being supervised, the model can employ rich features in the modeling process.

Mention-Ranking Models

Recasting event coreference as a classification task may not be a good idea, however. Recall that mention-pair models consider each candidate antecedent of an event mention to be resolved independently of other candidate antecedents. As a result, they can only determine how good a candidate antecedent is relative to the event mention, but not how good it is relative to other candidate antecedents. Ranking models address this weakness by allowing candidate antecedents of a mention to be ranked *simultaneously*. Motivated by their successful application to entity coreference resolution [Denis and Baldridge, 2008; Durrett and Klein, 2013], Lu and Ng [2017b] train a probabilistic *mention-ranking* model that ranks the candidate antecedents of an event mention so that its correct antecedent has the highest rank. Rather than train a model that maximizes the probability of selecting the correct antecedent for each event mention independently of each other, Lu and Ng train a model to select the antecedents for the event mentions in a document in a *collective* manner by having it assign the highest probability to the correct *vector* of antecedents given all the event mentions. Inference is easy: the most probable (i.e., highest-ranked) candidate antecedent of an event mention is selected to be its antecedent independently of other event mentions.

Easy-First Models

Easy-first models have been successfully applied to many NLP tasks, including entity coreference resolution [Lee *et al.*, 2013]. Easy-first coreference models operate in an iterative fashion, aiming to make easy linking decisions first and subsequently exploit these easy decisions (as additional knowledge) to make hard linking decisions.

One of the earliest event coreference resolvers that employs an easy-first approach is Stanford's resolver [Lee *et al.*, 2012]. This resolver iteratively bootstraps event coreference output using entity coreference output and vice versa. Specifically, it incrementally builds clusters of event and entity mentions. As clusters become larger, more information becomes available. To exploit the additional information, the features of both the event coreference resolver and the entity coreference model are regenerated.

Liu *et al.* [2014] attempt to improve the two-step "classify and cluster" approach described above by adding a third step,

where they keep propagating arguments from one mention in an event coreference cluster to another mention in the same cluster until all mentions in an event coreference cluster share the same arguments. This is an instance of the easy-first approach, as argument propagation helps to identify arguments for event mentions that are otherwise difficult to extract.

Lu and Ng [2016] implement an easy-first approach for event coreference resolution using six *sieves*. A sieve is composed of either a set of hand-crafted rules or a machine-learned classifier for classifying a subset of the mention pairs in a test document. Being an easy-first approach, the six sieves are arranged as a pipeline in decreasing order of precision. When two event mentions are posited as coreferent by a sieve, any argument extracted for one mention will be shared by the other mention. In addition, later sieves can exploit the event coreference decisions made by earlier sieves.

Choubey and Huang [2017] build a two-step agglomerative clustering algorithm for within- and cross-document coreference. In the first step, an iterative algorithm that alternates between within- and cross-document event coreference is used to merge within- or cross-document clusters by exploiting the merging decisions made in earlier iterations. Like Liu *et al.* [2014], the arguments of the event mentions in the same cluster are shared after each merge. When no more merging can be done, the algorithm proceeds to the second step where additional clusters are merged in an iterative fashion as follows. If the mentions in cluster c_1 are tightly associated (i.e., having the same dependency relations) or loosely associated (i.e., co-occurring in the same sentential context) with those in c_3, and the mentions in cluster c_2 are also tightly or loosely associated with those in c_3, then c_1 and c_2 will be merged.

Joint Models

The aforementioned models all adopt a pipeline architecture, where event triggers and arguments are extracted prior to event coreference resolution. Hence, errors from the upstream components (trigger identification and argument identification) will propagate to the event coreference resolver.

One solution to the error propagation problem is to employ *joint inference* over the outputs of different tasks in the IE pipeline. Chen and Ng [2016] perform joint inference via Integer Linear Programming (ILP) over the outputs of the models trained for the four key tasks in the IE pipeline, namely entity extraction, entity coreference, event extraction, and event coreference. Lu *et al.* [2016] perform joint inference using Markov Logic Networks (MLNs) over four tasks, namely trigger identification, argument extraction, entity coreference and event coreference. Joint inference allows an event coreference resolver and its upstream components to mutually influence (and possibly improve) each other by exploiting *background knowledge* expressed in the form of manually specified *constraints* on the tasks involved. One such constraint, for instance, could be that two triggers that do not have the same event subtype cannot be coreferent. While ILP is typically used to encode hard constraints, MLNs enables both soft and hard formulas to be encoded.

Another solution to the error propagation problem is *joint learning*. Araki and Mitamura [2015] formalize the task of jointly learning event trigger identification and event coreference resolution as a structured prediction problem that is learned using the structured perceptron training algorithm. They employ segment-based decoding with multiple-beam search for event trigger identification, and combine it with best-first clustering for event coreference resolution in document-level joint decoding. Lu and Ng [2017a] jointly learn event coreference resolution, event trigger detection, and event anaphoricity determination[3]. Motivated by Durrett and Klein's [2014] joint model for entity analysis, Lu and Ng build a structured conditional random field model with (1) unary factors, which encode the features specific for each task, and (2) higher-order factors, which capture the interactions between each pair of tasks in a soft manner. Each candidate event mention in a given document is associated with three output variables that encode its trigger subtype, its anaphoricity, and its antecedent. The goal is to learn which combination of values of these output variables are the most probable.

3.2 Semi-Supervised Models

Supervised models suffer from the data acquisition bottleneck, where manually annotating data for all the components in the IE pipeline is expensive. This is especially true for resource-scarce languages. To address this problem, researchers have employed *active learning* to select informative instances, showing that only a small number of training sentences need to be annotated to achieve state-of-the-art event coreference performance [Chen and Ng, 2016]. Another attempt is made to utilize large amounts of out of domain text data [Peng *et al.*, 2016]. The idea is to (1) represent event structures by five event semantic components, namely action, argument, time, location, and sentence/clause; (2) convert each event component to its corresponding vector representation using different methods, namely explicit semantic analysis, Brown cluster, Word2Vec and dependency-based word embedding; and (3) concatenate all components to form a structured vector representation. These semantic representations are induced from a data set that is not part of the existing annotated event collections and not even from the same domain. Finally, event coreference resolution can be recast as the task of comparing the similarities between event vectors.

3.3 Unsupervised Models

Unsupervised models are proposed to eliminate a model's reliance on annotated data. The vast majority of the existing unsupervised event coreference models are probabilistic generative models. Bejan and Harabagiu [2014] (B&H) propose several nonparametric Bayesian models for event coreference resolution that probabilistically infer event clusters both *within* a document and *across* multiple documents. One model uses the hierarchical Dirichlet process. It consists of a set of Dirichlet Processes (DPs), in which each DP is associated with each document, and each mixture component is an event coreference cluster shared across documents. This model has the advantage of automatically inferring the number of event clusters in a document. Despite this advantage,

[3]Event anaphoricity refers to the task of determining whether an event mention is the first mention in an event coreference chain.

the model has limitations in representing feature-rich objects. Consequently, B&H extend this model so that it can consider additional linguistic features derived from WordNet and FrameNet, for instance, instead of just representing each data point by its corresponding word. However, using a feature-rich representation may increase the complexity of a Bayesian model and there is no guarantee that all the features have a positive impact on the task. As a result, B&H extend their model with a feature selection mechanism that automatically selects a finite set of salient features. In addition, they propose another Bayesian model with a mechanism for capturing the structural dependencies between objects.

B&H show that their models that exploit the semantic information extracted from WordNet and FrameNet contribute to coreference performance significantly. However, the lack of comparable lexical knowledge bases complicates the design of event coreference resolvers in languages other than English. To address this problem, Chen and Ng [2015] design a probabilistic model whose parameters are estimated using EM for computing the probability that two event mentions are coreferent. Its generative process is not language-dependent and does not rely on features extracted from lexical knowledge bases, so it could be applied to languages where neither annotated data nor large-scale knowledge bases are available.

4 Linguistic Features

In this section, we give an overview of the linguistic features that have been used for event coreference resolution.

Lexical features explicitly or implicitly compare the event triggers of a pair of event mentions. Commonly used lexical features include (1) pair features such as the trigger pairs and the part-of-speech pairs of the two event triggers under consideration; (2) string-matching features such as exact string match, partial string match, stem match and lemma match; (3) trigger similarity features such as Dice coefficient, edit distance, Jaro coefficient, and the similarity computed based on term frequency vectors. In addition, the surrounding words of the event triggers and their similarities have been used as features. String-matching features have been shown to contribute significantly to the performance of an event coreference system [Lee *et al.*, 2012; Liu *et al.*, 2014; Chen and Ng, 2015; Yang *et al.*, 2015].

Argument features have also been exploited extensively for event coreference resolution, since event mentions having incompatible arguments are unlikely to be coreferent. Some argument features encode the number of overlapping arguments between two event mentions, the number of unique arguments that each event mention possesses, and whether the two event mentions have conflicting time and location arguments [Chen and Ji, 2009]. Other argument features encode the similarities between arguments, such as whether the two arguments are (entity-)coreferent, the surface similarities using Dice coefficient, and the WuPalmer WordNet similarity between argument head words [Liu *et al.*, 2014].

To extract argument features, an event argument extractor and an entity coreference resolver are typically needed. While *gold* arguments are used in early work on event coreference, [Chen and Ji, 2009; McConky *et al.*, 2012; Sangeetha and Arock, 2012], recent work focuses on building *end-to-end* resolvers using automatically extracted arguments. While some researchers train event argument extractors on event-annotated corpora to extract arguments and their roles that are specific to a given event ontology [Chen and Ng, 2014; Yang *et al.*, 2015], others extract event arguments and their roles heuristically from a PropBank-style semantic role labeler, which yields corpus-independent semantic roles such as ARG0, ARG1, etc. [Bejan and Harabagiu, 2014; Choubey and Huang, 2017]. For instance, ARG0 denotes an AGENT, a DOER or an ACTOR, whereas ARG1 denotes a PATIENT, a THEME or an EXPERIENCER.

Lee *et al.* [2012] and Yang *et al.* [2015] show that argument information, when encoded as features, is useful for event coreference resolution. However, event coreference resolvers can further be improved by improving existing argument extractors and entity coreference resolvers, since the precision and recall errors produced in these two modules limit the extent to which the resulting argument-based features contribute to event coreference performance.

Semantic features have been extracted from lexical-semantic resources (e.g., WordNet, FrameNet, VerbOcean), Brown clusters, and the word embeddings produced by Word2Vec for computing the similarity between two event mentions [Liu *et al.*, 2014; Yu *et al.*, 2016]. Experiments show that the embedding-based similarity feature has the highest weight among all features, hence suggesting its usefulness [Yang *et al.*, 2015; Choubey and Huang, 2017]. Event (sub)type match has also been shown to be a strong indicator of event coreference [Chen and Ng, 2014].

Finally, **discourse features** encode the token, event and sentence distance between two event mentions, as well as the position of an event mention in newswire articles [Liu *et al.*, 2014; Cybulska and Vossen, 2015]. The ablation experiments in Chen and Ng [2015] show that discourse features do not contribute to event coreference performance as much as other types of features, however.

5 Evaluation

In this section, we focus on two evaluation issues.

5.1 Extracting Candidate Event Mentions

Since an end-to-end event coreference resolver operates on the event mentions extracted by the event extraction component, it is conceivable that event coreference performance is significantly affected by event mention (i.e., trigger) detection performance, in pretty much the same way that entity coreference performance is affected by entity mention detection performance. Unfortunately, trigger detection is another challenging task that is far from being solved.

Given the difficulty of trigger detection, some researchers have chosen to work on the *non-end-to-end* version of the event coreference task, where they assume the existence of an oracle that provides the *gold* event mentions of a document to which they apply their event coreference algorithm. As we will see in the next section, using gold mentions yields results that are considerably better than using system (i.e., automatically extracted) mentions. This should not be surprising:

Language	Corpus	WD /CD	Approach	System	End-to-End?	Coreference						Trigger
						B^3	MUC	$CEAF_e$	BLANC	CoNLL	AVG	F
English	ECB+	WD	Easy-first	Choubey and Huang [2017]	Yes	72.40	62.60	71.80	–	68.93	–	–
		CD	Easy-first	Choubey and Huang [2017]	Yes	61.00	73.40	56.50	–	63.63	–	–
	ACE 2005	WD	Mention-pair	Peng et al. [2016]	Yes	59.90	47.10	58.70	44.40	–	52.53	69.10
		WD	Mention-pair	Peng et al. [2016]	No	92.80	74.90	87.10	83.80	–	84.70	gold
	KBP 2015	WD	Easy-First	Lu and Ng [2016]	Yes	45.39	44.07	38.67	33.14	–	40.32	57.45
	KBP 2016	WD	Joint	Lu and Ng [2017a]	Yes	40.90	27.41	39.00	25.00	–	33.08	49.30
	KBP 2017	WD	Easy-First	Jiang et al. [2017]	Yes	43.84	30.63	39.86	26.97	–	35.33	56.19
Chinese	ACE 2005	WD	Unsupervised	Chen and Ng [2015]	Yes	40.20	42.80	41.60	26.90	41.53	–	–
	KBP 2016	WD	Joint	Lu and Ng [2017a]	Yes	33.01	27.94	29.96	20.24	–	27.79	40.53
	KBP 2017	WD	Easy-First	Lu and Ng [2017c]	Yes	34.18	27.07	32.22	18.57	–	28.01	46.76
Spanish	KBP 2016	WD	Mention-pair	Yu et al. [2016]	Yes	22.05	19.04	18.56	12.43	–	18.02	37.23
	KBP 2017	WD	Mention-pair	Duncan et al. [2017]	Yes	9.90	3.89	10.39	2.04	–	6.55	23.25

Table 2: Performance of state-of-the-art event coreference resolvers on benchmark data sets.

coreference on gold mentions is a substantially simplified version of the event coreference task because system mentions typically significantly outnumber gold mentions. Some researchers argue that non-end-to-end coreference evaluations are unrealistic, as event mention extraction is an integral part of an end-to-end event coreference resolver.

5.2 Evaluation Metrics

The choice of coreference evaluation metrics is an issue that has been discussed extensively in the coreference research community for many years. Since researchers cannot agree on which evaluation metric is the best to use, multiple metrics are typically used to evaluate an event coreference resolver, largely following the evaluation setup that was standardized in the CoNLL 2011/2012 shared tasks on entity coreference resolution. More specifically, four metrics that are originally developed for entity coreference evaluation are commonly used to evaluate event coreference resolvers, namely the link-based MUC metric [Vilain *et al.*, 1995], the mention-based B^3 metric [Bagga and Baldwin, 1998], the entity-based $CEAF_e$ metric [Luo, 2005] and the Rand index-based BLANC metric [Recasens and Hovy, 2011]. It is also common to report the CoNLL score, which is the unweighted average of the F-scores produced by the first three metrics, and the AVG score, which is the unweighted average of the F-scores produced by all four metrics.

6 The State of the Art

In this section, we provide an overview of the performance of state-of-the-art systems on benchmark data sets.

Table 2 shows the best results to date on each data set. Coreference results are reported in terms of F-score obtained using four metrics, namely MUC, B^3, $CEAF_e$ and BLANC, as well as CoNLL and AVG. Trigger performance is reported in terms of F-score, where an event mention is considered correctly detected if it has an exact match with a gold mention in terms of boundary, event type, and event subtype.

Several points deserve mention. First, both event coreference and trigger detection are far from being solved. Specifically, the best trigger detection result is around 69.1 F-score. The best coreference result is even worse: it is only around 68.9 (w.r.t. CoNLL) and 52.5 (w.r.t. AVG). Second, as can be seen from Peng et al.'s [2016] results, the AVG score drops precipitously from 84.7 to 52.5 when gold event mentions are replaced with system event mentions. These results corroborate our earlier claim that using gold mentions for event coreference resolution substantially simplifies the task. Finally, the best coreference results are achieved for English, and the worst results are achieved for Spanish.

7 Concluding Remarks

While researchers are making continued progress on the event coreference task despite its difficulty, a natural question is: what are the promising directions for future work?

Given recent successes on applying joint learning to event coreference resolution, it may be worthwhile to investigate joint models further. For instance, while previous work has applied joint *inference* to the four key tasks in IE (entity extraction, entity coreference, event extraction, and event coreference), one can determine if it is possible to jointly *learn* these four tasks. Jointly learning four tasks is extremely challenging owing to the computational complexity involved, so novel scalable learning algorithms will be needed.

If joint modeling is not possible (e.g., because annotated data is not sufficient for training models for the related tasks), we may need to employ sophisticated features to improve state-of-the-art resolvers despite the difficulty in extracting/inducing such features. Given recent successes on employing word vectors for event coreference resolution [Choubey and Huang, 2017], one can take this idea further and learn representations from complex features, including those that are derived from automatically computed arguments and entity coreference chains.

Event coreference models cannot be applied to the vast majority of the world's low-resource languages for which event coreference-annotated data is not readily available. It would be interesting to examine whether there are language-specific issues that could affect the effective application of unsupervised, semi-supervised, and annotation projection approaches to event coreference resolution involving less-studied languages. In addition, if large lexical knowledge bases do not exist for the target language, it would be important to investigate alternative methods for obtaining semantic knowledge.

Finally, there are other types of event coreference that are less studied than the full event coreference task we examined in this paper. One is partial coreference. Hovy *et al.*

[2013] define two types of partial event coreference relations: subevent relations and membership relations. Subevent relations form a stereotypical sequence of events, whereas membership relations represent instances of an event collection. We refer the reader to Araki *et al.* [2014] for details.

Acknowledgments

We thank the two anonymous reviewers for their detailed comments on an earlier draft of the paper. This work was supported in part by NSF Grants IIS-1219142 and IIS-1528037.

References

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

[Allan *et al.*, 1998] James Allan, Jaime G. Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic detection and tracking pilot study final report. In *Proceedings of the Broadcast News Understanding and Transcription Workshop*, pages 194–218, 1998.

[Araki and Mitamura, 2015] Jun Araki and Teruko Mitamura. 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, 2015.

[Araki *et al.*, 2014] Jun Araki, Eduard Hovy, and Teruko Mitamura. Evaluation for partial event coreference. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 68–76, 2014.

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

[Bejan and Harabagiu, 2008] Cosmin Adrian Bejan and Sanda Harabagiu. A linguistic resource for discovering event structures and resolving event coreference. In *Proceedings of the Sixth Language Resources and Evaluation Conference*, pages 2881–2887, 2008.

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

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

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

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

[Chen and Ng, 2016] Chen Chen and Vincent Ng. Joint inference over a lightly supervised information extraction pipeline: Towards event coreference resolution for resource-scarce languages. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 2913–2920, 2016.

[Chen *et al.*, 2009] Zheng Chen, Heng Ji, and Robert Haralick. A pairwise event coreference model, feature impact and evaluation for event coreference resolution. In *Proceedings of the Workshop on Events in Emerging Text Types*, number 3, pages 17–22, 2009.

[Chen *et al.*, 2011] Bin Chen, Jian Su, Sinno Jialin Pan, and Chew Lim Tan. A unified event coreference resolution by integrating multiple resolvers. In *Proceedings of the Fifth International Joint Conference on Natural Language Processing*, pages 102–110, 2011.

[Choubey and Huang, 2017] Prafulla Kumar Choubey and Ruihong Huang. Event coreference resolution by iteratively unfolding inter-dependencies among events. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2124–2133, 2017.

[Cybulska and Vossen, 2014] Agata Cybulska and Piek Vossen. Using a sledgehammer to crack a nut? Lexical diversity and event coreference resolution. In *Proceedings of the Ninth Language Resources and Evaluation Conference*, pages 4545–4552, 2014.

[Cybulska and Vossen, 2015] Agata Cybulska and Piek Vossen. Translating granularity of event slots into features for event coreference resolution. In *Proceedings of the Third Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 1–10, 2015.

[De Marneffe *et al.*, 2008] Marie-Catherine De Marneffe, Anna N. Rafferty, and Christopher D. Manning. Finding contradictions in text. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1039–1047, 2008.

[Denis and Baldridge, 2008] Pascal Denis and Jason Baldridge. Specialized models and ranking for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 660–669, 2008.

[Duncan *et al.*, 2017] Chase Duncan, Liang-Wei Chan, Haoruo Peng, Hao Wu, Shyam Upadhyay, Nitish Gupta, Chen-Tse Tsai, Mark Sammons, and Dan Roth. UI CCG TAC-KBP2017 submissions: Entity discovery and linking, and event nugget detection and co-reference. In *Proceedings of the Text Analysis Conference*, 2017.

[Durrett and Klein, 2013] Greg Durrett and Dan Klein. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1982, 2013.

[Durrett and Klein, 2014] Greg Durrett and Dan Klein. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490, 2014.

[Hovy *et al.*, 2013] Eduard Hovy, Teruko Mitamura, Felisa Verdejo ETSI Informática, Juan del Rosal, Jun Araki, and Andrew Philpot. Events are not simple: Identity, non-identity, and quasi-identity. In *Proceedings of the the First Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 21–28, 2013.

[Humphreys *et al.*, 1997] Kevin Humphreys, Robert Gaizauskas, and Saliha Azzam. Event coreference for information extraction. In *Proceedings of the ACL/EACL Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 75–81, 1997.

[Ji and Grishman, 2011] Heng Ji and Ralph Grishman. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Com-*

putational Linguistics: Human Language Technologies, pages 1148–1158, 2011.

[Jiang *et al.*, 2017] Shanshan Jiang, Yihan Li, Tianyi Qin, Qian Meng, and Bin Dong. SRCB entity discovery and linking (EDL) and event nugget systems for TAC 2017. In *Proceedings of the Text Analysis Conference*, 2017.

[LDC, 2005] LDC. ACE (Automatic Content Extraction) English annotation guidelines for events. Technical report, Linguistic Data Consortium, 2005.

[Lee *et al.*, 2012] Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 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, 2012.

[Lee *et al.*, 2013] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916, 2013.

[Li *et al.*, 2006] Wenjie Li, Mingli Wu, Qin Lu, Wei Xu, and Chunfa Yuan. Extractive summarization using inter-and intra-event relevance. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 369–376, 2006.

[Liu *et al.*, 2014] Zhengzhong Liu, Jun Araki, Eduard Hovy, and Teruko Mitamura. Supervised within-document event coreference using information propagation. In *Proceedings of the Ninth Language Resources and Evaluation Conference*, pages 4539–4544, 2014.

[Lu and Ng, 2016] Jing Lu and Vincent Ng. Event coreference resolution with multi-pass sieves. In *Proceedings of the 10th Language Resources and Evaluation Conference*, pages 3996–4003, 2016.

[Lu and Ng, 2017a] Jing Lu and Vincent Ng. Joint learning for event coreference resolution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 90–101, 2017.

[Lu and Ng, 2017b] Jing Lu and Vincent Ng. Learning antecedent structures for event coreference resolution. In *Proceedings of the 16th IEEE International Conference on Machine Learning and Applications*, pages 113–118, 2017.

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

[Lu *et al.*, 2016] Jing Lu, Deepak Venugopal, Vibhav Gogate, and Vincent Ng. Joint inference for event coreference resolution. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 3264–3275, 2016.

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

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

[MUC6, 1995] MUC-6. Proceedings of the Sixth Message Understanding Conference. 1995.

[MUC7, 1998] MUC-7. Proceedings of the Seventh Message Understanding Conference. 1998.

[Narayanan and Harabagiu, 2004] Srini Narayanan and Sanda Harabagiu. Question answering based on semantic structures. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 693–701, 2004.

[Ng and Cardie, 2002] Vincent Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111, 2002.

[Nguyen *et al.*, 2016] Thien Huu Nguyen, Adam Meyers, and Ralph Grishman. New York University 2016 system for KBP event nugget: A deep learning approach. In *Proceedings of the Text Analysis Conference*, 2016.

[Peng *et al.*, 2016] Haoruo Peng, Yangqi Song, and Dan Roth. Event detection and co-reference with minimal supervision. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 392–402, 2016.

[Pradhan *et al.*, 2007] Sameer S. Pradhan, Lance Ramshaw, Ralph Weischedel, Jessica MacBride, and Linnea Micciulla. Unrestricted coreference: Identifying entities and events in OntoNotes. In *Proceedings of the First IEEE International Conference on Semantic Computing*, pages 446–453, 2007.

[Pustejovsky *et al.*, 2003] James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. TimeML: Robust specification of event and temporal expressions in text. In *Proceedings of the Fifth International Workshop on Computational Semantics*, 2003.

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

[Sangeetha and Arock, 2012] Satyan Sangeetha and Michael Arock. Event coreference resolution using mincut based graph clustering. In *Proceedings of the Fourth International Workshop on Computer Networks & Communications*, pages 253–260, 2012.

[Song *et al.*, 2015] Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. From Light to Rich ERE: Annotation of entities, relations, and events. In *Proceedings of the Third Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98, 2015.

[Soon *et al.*, 2001] Wee Meng Soon, Hwee Tou Ng, Daniel Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.

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

[Yang *et al.*, 2015] Bishan Yang, Claire Cardie, and Peter Frazier. A hierarchical distance-dependent Bayesian model for event coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:517–528, 2015.

[Yu *et al.*, 2016] Dian Yu, Xiaoman Pan, Boliang Zhang, Lifu Huang, Di Lu, Spencer Whitehead, and Heng Ji. RPI BLENDER TAC-KBP2016 system description. In *Proceedings of the Text Analysis Conference*, 2016.