A Pairwise Event Coreference Model, Feature Impact and Evaluation for Event Coreference Resolution

Zheng Chen
The Graduate Center
The City University of New York
zchen1@gc.cuny.edu

Heng Ji
Queens College and The Graduate Center
The City University of New York
hengji@cs.qc.cuny.edu

Robert Haralick
The Graduate Center
The City University of New York
haralick@aim.com

Abstract

In past years, there has been substantial work on the problem of entity coreference resolution whereas much less attention has been paid to event coreference resolution. Starting with some motivating examples, we formally state the problem of event coreference resolution in the ACE [1] program, present an agglomerative clustering algorithm for the task, explore the feature impact in the event coreference model and compare three evaluation metrics that were previously adopted in entity coreference resolution: MUC F-Measure, B-Cubed F-Measure and ECM F-Measure.

Keywords

Pairwise Event Coreference Model, Event Coreference Resolution, Event Attribute

1. Introduction

In this paper, we address the task of event coreference resolution specified in the Automatic Content Extraction (ACE) program: grouping all the mentions of events in a document into equivalent classes so that all the mentions in a given class refer to a unified event. We adopt the following terminologies used in ACE [1]:

- Entity: an object or set of objects in the world, such as person, organization, facility.
- Event: a specific occurrence involving participants.
- Event trigger: the word that most clearly expresses an event's occurrence.
- Event argument: an *entity*, or a *temporal expression* or a *value* that has a certain role (e.g., PLACE) in an event.
- Event mention: a sentence or phrase that mentions an event, including a distinguished trigger and involving arguments. An event is a cluster of event mentions.
- Event attributes: an event has six event attributes, event type, subtype, polarity, modality, genericity, and tense.

We demonstrate some motivating examples in table 1 (event triggers are surrounded by curly brackets and event arguments are underlined).

In example 1,event mention *EM1* corefers with *EM2* because they have the same event type and subtype (CONFLICT: ATTACK) indicated by two verb triggers "tore" and "exploded" respectively, and the argument "a waiting shed" in *EM1* corefers with "the waiting shed" in *EM2*. In example 2, *EM1*, *EM2* and *EM3* corefer with each other because they have the same event type and subtype (LIFE:MARRY) indicated by a verb trigger "wed" and two noun triggers "ceremony" and "nuptials" respectively. Furthermore, the two persons "Rudolph Giuliani" and "Judith Nathan" involving in the "Marry" event in *EM1* corefer with "Giuliani" and "Nathan" in *EM3* respectively. In example 3, *EM1* does not corefer with *EM2* although they have the same event type and subtype (LIFE:DIE) because the event attribute "polarity" of *EM1* is "POSITIVE" (occurred) while in *EM2*, it is "NEGATIVE" (not occurred).

Table 1. Motivating examples for event coreference resolution

Example1 *EM1*: A powerful bomb {tore} through a waiting shed at the Davao City international airport. *EM2*: The waiting shed literally {exploded}.
Example2 *EM1*: Rudolph Giuliani will {wed} his companion, Judith Nathan, on May 24 in the ex-mayor's old home. *EM2*: Mayor Michael Bloomberg, will perform the {ceremony}. *EM3*: The Giuliani-Nathan {nuptials} will be a first for Bloomberg, who is making an exception from his policy of not performing weddings.
Example3 *EM1*: At least 19 people were {killed} in the first blast. *EM2*: There were no reports of {deaths} in the second blast.

The major contributions of this paper are:

(1) A formal statement of event coreference resolution and an algorithm for the task.

(2) A close study of four event attributes: polarity, modality, genericity and tense.

(3) A close study of feature impact on the performance of the pairwise event coreference model.

[1] http://www.nist.gov/speech/tests/ace/

2. Event Coreference Resolution

We formulate the problem of event coreference resolution as an agglomerative clustering task. The basic idea is to start with singleton event mentions, traverse through each event mention (from left to right) and iteratively merge the active event mention into a prior established event or start the event mention as a new event. We first introduce the notation needed for our algorithm.

2.1 Notation

Let I be the set of positive integers. Let A be a set of attributes and V be a set of values. Some attributes may have no values and some attributes may have one or more values. Any information about an event is a subset of $A \times V$, and the same applies to an event mention. Such abstraction makes it possible for us to extend the meaning of attributes.

In this paper we state that an event includes the following attribute members: 6 event attributes (type, subtype, modality, polarity, genericity and tense), a set of arguments, and a set of event mentions. Accordingly, we have the following notation:

Let e be an ACE event. Let $e.arg$ be the set of arguments (*entities, temporal expressions and values*) in the event e. Let $e.ems$ be the set of event mentions in the event e.

An event mention has a distinguished trigger and a set of arguments. Accordingly, we have the following notation:

Let em be an event mention. Let $em.trigger$ be the event trigger. Let $em.arg$ be the set of arguments (*entities, temporal expressions and values*) in the event mention em.

Let M be the set of possible event mentions in a document D. Let $< em_i \in M \mid i = 1, \ldots, N >$ be the N event mentions in the document D listed in the order in which they occur in the document.

Let E be the set of possible events in the document D. Let $< e_j \in E \mid j = 1, \ldots, K >$ be the K events.

The goal of event coreference resolution is to construct a function $f: I \to I$, mapping event mention index $i \in I$ to event index $j \in I$.

Initially, each event mention em is wrapped in an event $e^{'}$ so that $e^{'}$ contains a single event mention em. We denote the wrapping function as $\alpha: M \to E'$, i.e., $e^{'} = \alpha(em)$where E' is the set of $e^{'}$. Furthermore, em and $e^{'}$ satisfy the following properties: (1) $e^{'}.ems = em$ (2) $e^{'}.arg = em.arg$

2.2 Algorithm

We describe an agglomerative clustering algorithm that gradually builds up the set of events by scanning each event mention from left to right.

Let E_0 be the initial set of established events and $E_0 = \emptyset$. $E_1 = \{\alpha(em_1)\}$ and $f(1) = 1$. Let δ be a threshold. Let $coref: E \times M \to (0,1)$ be a function which gives a score to any (event, event mention) pair.

At each iteration k ($k = 2, \ldots, N$), let $e_j \in E_{k-1}$ satisfy

$coref(e_j, em_k) \geq coref(e, em_k)$ for any $e \in E_{k-1}$

If $coref(e_j, em_k) \geq \delta$, then $f(k) = j$ and

$$E_k = \{e_1^k, \ldots, e_{N_k}^k\}$$

where $e_n^k = e_n^{k-1}$for $n \neq j$ and $e_n^k.ems = e_n^{k-1}.ems \cup \{em_k\}, e_n^k.arg = e_n^{k-1}.arg \cup em_k.arg$ for $n = j$.

If $coref(e_j, em_k) < \delta$, then $f(k) = N_{k-1} + 1$ and

$$E_k = E_{k-1} \cup \{e_{N_k}^k\}$$

where $N_k = N_{k-1} + 1$ and $e_{N_k}^k = \alpha(em_k)$

After $N - 1$ iterations, we resolve all the event coreferences in the document.

The complexity of the algorithm is $O(N^2)$. However, if we only consider those event mentions with the same event type and subtype, we can decrease its running time.

2.3 Pairwise Event Coreference Model

A key issue in the above algorithm is how to compute the coreference function $coref(\cdot, \cdot)$ which indicates the coreference score between the active event mention and a prior established event. We construct a Maximum-entropy model for learning such function. The features applied in our model are tabulated in Table 2. We categorize our features into *base*, *distance*, *arguments* and *attributes* feature sets to capture trigger relatedness, trigger distance, argument compatibility and event attribute compatibility respectively.

In this paper, we run NYU's 2005 ACE system [2] to tag event mentions. However, their system can only extract triggers, arguments and two event attributes (event type and subtype) and cannot extract the other four event attributes. Therefore, we developed individual components for those four event attributes (polarity, modality, genericity and tense). Such efforts have been largely neglected in the prior research due to their low weights in the ACE scoring metric [1]. The event attributes absolutely play an important role in event coreference resolution because two event mentions cannot corefer with each other if any of the attributes conflict with each other. We encode the event attributes as features in our model and study their impact on the system performance. In the next section, we describe the four event attributes in details.

3. Extracting the Four Event Attributes

3.1 Polarity

An event is NEGATIVE if it is explicitly indicated that the event did not occur, otherwise, the event is POSITIVE. The following list reviews some common ways in which NEGATIVE polarity may be expressed (triggers are

Table 2. Feature categories for the pairwise event coreference model

Category	**Features**	**Feature Values** (aem: the active event mention, e: a partially-established event, lem: the last event mention in e)
Base	type_subtype	pair of event type and subtype in aem
	nominal	1 if the trigger of aem is nominal
	nom_number	plural or singular if the trigger of aem is nominal
	pronominal	1 if the trigger of aem is pronominal
	exact_match	1 if the trigger spelling of aem matches the trigger spelling of an event mention in e
	stem_match	1 if the trigger stem in aem matches the trigger stem of an event mention in e
	trigger_sim	the maximum of quantized semantic similarity scores (0-5) using WordNet resource among the trigger pairs of aem and an event mention in e
	trigger_pair	trigger pair of aem and lem
	pos_pair	part-of-speech pair of triggers of aem and lem
Distance	token_dist	how many tokens between triggers of aem and lem (quantized)
	sentence_dist	how many sentences aem and lem are apart (quantized)
	event_dist	how many events in between aem and lem (quantized)
Arguments	overlap_num, overlap_roles	overlap number of arguments and their roles (role and id exactly match) between aem and e
	prior_num, prior_roles	the number of arguments that only appear in e and their roles
	act_num, act_roles	the number of arguments that only appear in aem and their roles
	coref_num	the number of arguments that corefer with each other but have different roles between aem and e
	time_conflict	1 if both aem and e have an argument with role "Time-Within" and their values conflict
	place_conflict	1 if both aem and e have an argument with role "Place" and their values conflict
Attributes	mod,pol,gen, ten	four event attributes in aem: modality, polarity, genericity, and tense
	mod_conflict, pol_conflict, gen_conflict, ten_conflict	four boolean values indicating whether the attributes of aem and e conflict

surrounded by curly brackets, the words indicating NEGATIVE are underscored)

- Using a negative word such as not, no

Guns don't {kill} people, people do.

No death sentence has ever been {executed} in the country.

- Using context, e.g., the embedding predicate with a negative meaning or sentence patterns

Bush indefinitely postponed a {visit} to Canada.

She had decided to stay home rather than {go} to a dance.

3.2 Modality

An event is ASSERTED if it is mentioned as if it were a real occurrence, otherwise it is OTHER. Two "ASSERTED" examples are listed as follows:

At least 19 people were {killed} in Tuesday's blast.

We condemn all {attacks} against civilians in Haifa.

The "OTHER" examples have much more varieties. The examples include, but are not limited to (triggers are surrounded by curly brackets, the words indicating modality are underscored)

- believed events

I believe he will be {sentenced}.

- hypothetical events

If convicted of the killings, Vang {faces} life in prison.

- commanded and requested events

He was commanded to {leave} his country.

- threatened, proposed and discussed events

He was threatened to {pay} the ransom.

- desired events

He desires to be {elected}.

- promised events

The terrorist said he would {attack} the village.

The modality of events can be characterized by a veridicality axis that ranges from truly factual to counter-factual and a spectrum of modal types fall between the two extremes, expressing *degrees of possibility*, *belief*, *evidentiality*, *expectation*, *attempting*, and *command* [3]. Actually, ACE has largely simplified the problem, i.e., the modality is "ASSERTED" for the two extremes, and is "OTHER" for all the other modal types.

3.3 Genericity

An event is SPECIFIC if it is a single occurrence at a particular place and time, or a finite set of such occurrences; otherwise, it is GENERIC.

Some GENERIC examples are listed as follows:

Hamas vowed to continue its {attacks}.

Roh has said any pre-emptive {strike} against the North's nuclear facilities could prove disastrous.

3.4 Tense

The tense of events can be characterized by a temporal axis in which we define the time of publication or broadcast as the *textual anchor time*. The PAST events occurred prior to the anchor time; the FUTURE events have not yet occurred at the anchor time; the PRESENT events occur at the anchor time; all the other events are UNSPECIFIED.

3.5 Models for the Four Event Attributes

We construct a Maximum-entropy model for each of the four event attributes. All the models apply the following common features:

- the trigger and its part-of-speech
- event type and subtype
- the left two words of the trigger (lower case) and their POS tags
- the right two words of the trigger (lower case) and their POS tags

Furthermore, the polarity model also applies the following two features:

- the embedding verb of the trigger if any
- a boolean feature indicating whether a negative word exists (not, no, cannot or a word ending with n't) ahead of the trigger and within the clause containing the trigger.

The modality model also applies the following feature:

- a boolean feature indicating whether a modal auxiliary (may, can, etc.) or modal adverbs (possibly, certainly, etc.) exists ahead of the trigger and within the clause containing the trigger.

The genericity model also applies the following three features:

- a boolean feature indicating whether the event mention has a "PLACE" argument
- a boolean feature indicating whether the event mention has a "TIME-WITHIN" argument
- the number of arguments that the event mention has except "PLACE" and "TIME-WITHIN"

The tense model also applies the following two features:

- the first verb within the clause containing the trigger and its POS tag
- the head words of the "TIME-WITHIN" argument if the event mention has one

4. Experiments and Results

4.1 Data and Evaluation Metrics

For our experiments, we used the ACE 2005 English corpus which contains 599 documents in six genres: newswire, broadcast news, broadcast conversations, weblogs, newsgroups and conversational telephone speech transcripts. We first investigated the performance of the four event attribute classification models using the ground truth event mentions and system generated event mentions respectively. The evaluation metrics we adopted in this set of experiments are Precision (P), Recall (R) and F-Measure (F). We then validated our agglomerative clustering algorithm for the event coreference resolution using the ground truth event mentions and system generated event mentions respectively. The evaluation metrics we adopted in this set of experiments are three conventional metrics for entity coreference resolution, namely, MUC F-Measure [4], B-Cubed F-Measure [5] and ECM F-Measure [6]. We conducted all the experiments by ten times ten-fold cross validation and measured significance with the Wilcoxon signed rank test.

4.2 Performance of the Four Event Attribute Classification Models

Table 3 shows that the majority of event mentions are POSITIVE (5162/5349=0.965), ASSERTED (4002/5349 =0.748), SPECIFIC (4145/5349=0.775) and PAST (2720/5349=0.509).

Table 3. Statistics of the four event attributes in the corpus

Attribute	Instance counts in the ACE corpus
Polarity	NEGATIVE=187, POSITIVE=5162
Modality	ASSERTED=4002, OTHER=1347
Genericity	GENERIC=1204, SPECIFIC=4145
Tense	FUTURE=593,PAST=2720, PRESENT=152, UNSPECIFIED=1884

Table 4 shows the performance of the four event attribute classification models using the ground truth event mentions (perfect) and the system generated event mentions (system). For comparison, we also set up a

Table 4. Performance of the four event attribute classification models

	Polarity			Modality			Genericity			Tense		
	P	R	F	P	R	F	P	R	F	P	R	F
Perfect (majority)	0.966	1.0	0.983	0.748	1.0	0.856	0.777	1.0	0.874	0.510	1.0	0.675
Perfect (model)	0.968	1.0	0.984	0.784	1.0	0.879	0.795	1.0	0.885	0.644	1.0	0.783
System (majority)	0.969	0.573	0.720	0.779	0.519	0.622	0.792	0.523	0.629	0.550	0.432	0.483
System (model)	0.974	0.574	0.722	0.805	0.527	0.637	0.799	0.525	0.633	0.677	0.484	0.564

baseline for each case using the majority value as output (e.g., for Polarity attribute, we always set the value to POSITIVE because POSITIVE is the majority).

Table 4 shows that the improvements for Polarity, Modality and Genericity over the baselines are quite limited while the improvements for Tense are significant, either using ground truth event mentions or using system generated event mentions.

4.3 Determining Coreference Threshold δ

In order to determine the best coreference threshold δ in our agglomerative clustering algorithm, we conducted this set of experiments by integrating full feature sets (as listed in Table 2) in the pairwise event coreference model. We investigate how the performance varies by adjusting the coreference threshold δ. For this set of experiments, we use ground truth event mentions.

Figure 1 shows the F-scores based on the three evaluation metrics by varying the coreference threshold δ. The best MUC F-score, B-Cubed F-score and ECM F-score are obtained at $\delta = 0.25, \delta = 0.3, \delta = 0.3$ respectively. It is worth noting that the MUC F-score drops dramatically after $\delta = 0.5$. We observed that as the threshold increases, more singleton events are produced and the dramatic decrease in MUC recall cannot offset the increase in MUC precision. As [5], [6] have pointed out, MUC metric does not give any credit for separating out singletons, therefore it is not quite effective in evaluating system responses with many singletons. The B-Cubed curve shows similar fluctuations compared to the ECM curve.

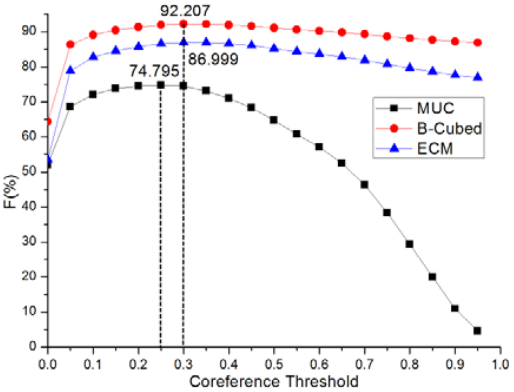

Figure 1. Determining the best coreference threshold δ

4.4 Feature Impact

Table 5 presents the impact of aggregating feature sets on the performance of our pairwise event coreference model using the ground truth event mentions (coreference threshold $\delta = 0.3$).

Table 5. Feature impact using ground truth event mentions

	MUC F	B-Cubed F	ECM F
Base	0.386	0.868	0.777
+Distance	0.446	0.866	0.781
+Arguments	0.530	0.879	0.804
+Attributes	0.723	0.919	0.865

Our Wilcoxon signed rank tests show that the F-score improvements are significant for all three metrics when we apply richer features except that there is a little deterioration for the distance feature set using B-Cubed metric. We observe that the improvement is dramatic using the MUC metric. However, it is not quite reasonable since we evaluate on the same system responses, varying in metrics. Since ECM overcomes some shortcomings of MUC and B-Cubed metrics as explained in [6], we focus on analyzing the results from ECM metric. In this setting, distance feature set contributes about 0.4% F-score improvement, while arguments feature set contributes nearly 2.4% F-score improvement. It is clear that the attribute feature set contributes the most significant contribution (6.08% absolute improvement).

We then investigate whether the feature sets have similar impacts on the pairwise event coreference model using the system generated event mentions.

Table 6. Feature impact using system generated event mentions

	MUC F	B-Cubed F	ECM F
Base	0.265	0.558	0.489
+Distance	0.254	0.548	0.483
+Arguments	0.274	0.552	0.490
+Attributes	0.28	0.554	0.492

Table 6 shows that the aggregated features do not bring great improvements using the system generated event mentions. The reason is that the spurious and missing event mentions labeled by the event extractor not only directly affect the final score of event coreference, but also lead to the deteriorated event coreference model which is learned from spurious feature values. We name the spurious event coreference caused by the spurious event mentions as *type I error*, the spurious event coreference caused by the model and the clustering algorithm as *type II error*. Similarly, we define *type I miss* and *type II miss*, one is caused by the missing event mentions and the other is caused by the model and the algorithm. Table 7 shows the average ratio of *type I error* and *type II error*, ratio of *type I miss* and *type II miss* for each model which only applies the features in each feature category. It is clear that the performance bottleneck of event coreference resolution comes from the performance of system event mentions.

Table 7. Ratio of *type I,II error* and ratio of *type I,II miss*

	type I error Vs. *type II error*	*type I miss* Vs. *type II miss*
Base	90%/10%	82.3%/17.7%
Distance	99.8%/0.2%	77.5%/22.5%
Arguments	95.2%/4.8%	81%/19%
Attributes	92.3%/7.7%	79.6%/20.4%

5. Related Work

Earlier work on event coreference (e.g. [7], [8]) in MUC was limited to several scenarios, e.g., terrorist attacks, management succession, resignation. The ACE program takes a further step towards processing more fine-grained events. Ahn presented an event extraction system in which event coreference resolver is located at the end of a pipeline of event extraction [9]. However, Ahn did not point out what evaluation metric he used. [10] presented a graph-based method for event coreference resolution and proposed two methods for computing the coreference score between two event mentions. However, they only reported evaluation results on ground-truth event mentions.

Our experiments show that a high-performance event coreference resolver relies on a high-performance event mention extractor. Earlier work on event extraction systems was presented in [9],[11],[12],[13],[14].

6. Conclusions and Future Work

We have formally stated the problem of event coreference resolution, presented an algorithm involving a pairwise event coreference model and studied the feature impacts on the pairwise event coreference model.

In the future, we will continue to put great efforts on improving the performance of event extraction system including trigger labelling, argument labelling and event attribute labelling. We believe that the improved components will finally help us improve the performance of event coreference resolution.

7. Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency under Contract No. HR0011-06-C-0023 via 27-001022, Google Research, CUNY Research Enhancement Program and GRTI Program. Any opinions, findings and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the U. S. Government.

8. References

[1] NIST. 2005. The ACE 2005 Evaluation Plan. http://www.itl.nist.gov/iad/mig/tests/ace/ace05/doc/ace05-evalplan.v3.pdf.

[2] R. Grishman, D. Westbrook, and A. Meyers. 2005. NYU's English ACE 2005 System Description. In *ACE 05 Evaluation Workshop*, Gaithersburg, MD.

[3] R. Saurí, M. Verhagen and J. Pustejovsky. 2006. Annotating and Recognizing Event Modality in Text. In *Proceedings of the 19th International FLAIRS Conference, FLAIRS 2006*. Melbourne Beach, Florida. May 11-13, 2006.

[4] M. Vilain, J. Burger, J. Aberdeen, D. Connolly and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*.

[5] A. Bagga and B. Baldwin. 1998. Algorithms for scoring coreference chains. *Proc. The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*.

[6] X. Luo. 2005. On coreference resolution performance metrics. *Proc. of HLT-EMNLP*.

[7] A. Bagga and B. Baldwin. 1999. Cross-document event coreference: Annotations, experiments, and observations. In *Proc. ACL-99 Workshop on Coreference and Its Applications*.

[8] K. Humphreys, R. Gaizauskas, S. Azzam. 1997. Event coreference for information extraction. In *Proceedings of the ACL* Workshop on Operational Factors in Practical Robust Anaphora Resolution for Unrestricted Texts.

[9] D. Ahn. 2006. The stages of event extraction. *Proc. COLING/ACL 2006 Workshop on Annotating and Reasoning about Time and Events*. Sydney, Australia.

[10] Z. Chen and H. Ji. 2009. Graph-based Event Coreference Resolution. *Proc. ACL-IJCNLP 2009 workshop on TextGraphs-4: Graph-based Methods for Natural Language Processing*.

[11] R. Grishman, D. Westbrook and A. Meyers. 2005. NYU's English ACE 2005 System Description. *Proc. ACE 2005 Evaluation Workshop*. Washington, US.

[12] H. Ji and R. Grishman. 2008. Refining Event Extraction Through Cross-document Inference. *Proc. ACL 2008*. Ohio, USA.

[13] Z. Chen and H. Ji. 2009. Language Specific Issue and Feature Exploration in Chinese Event Extraction. *Proc. HLT-NAACL 2009*. Boulder, Co.

[14] Z. Chen and H. Ji. 2009. Can One Language Bootstrap the Other: A Case Study on Event Extraction. *Proc. HLT-NAACL Workshop on Semi-supervised Learning for Natural Language Processing*. Boulder, Co.