Event Coreference Resolution: Algorithm, Feature Impact and Evaluation

Zheng Chen **Heng Ji**

Department of Computer Science

Queens College and The Graduate Center, The City University of New York

zchen1@gc.cuny.edu hengji@cs.qc.cuny.edu

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 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 equivalence 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, type, subtype, polarity, modality, genericity, and tense. ACE defines 8 event types (e.g., LIFE, MOVEMENT, TRANSACTION, etc.) with 33 event subtypes (e.g., the event type LIFE has 5 event subtypes BE-BORN, MARRY, DIVORCE, INJURE and DIE). The modality has a value of "ASSERTED" if an event is a real occurrence, otherwise, it has a value of "OTHER". The polarity has a value of "NEGATIVE" if an event did not occur, otherwise, it has a value of "POSITIVE". The genericity has a value of SPECIFIC if an event is a singular occurrence at a particular place and time, otherwise, it has a value of "GENERIC". The tense value is determined with respective to the speaker or author, and the possible values are PAST, FUTURE, PRESENT and UNSPECIFIED.

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

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*: Major US insurance group AIG is in the final stage of talks to {take over} General Electric's Japanese life insurance arm. *EM2*: The total {purchase} cost is estimated at 300 billion yen. *EM3*: AIG is also likely to {take over} GE's US car and fire insurance operations.
Example4 *EM1*: At least 19 people were {killed} in the first blast. *EM2*: There were no reports of {deaths} in the second blast.

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

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* corefers with *EM2* (assuming *EM2* immediately follows *EM1* in the document) because they have the same event type and subtype (TRANSACTION: TRANSFER-OWNERSHIP), but *EM1* does not corefer with *EM3* because the seller “…arm” in *EM1* is different from “…operations” in *EM3* although “AIG” is the common buyer. In example 4, *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).

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 impactss on the performance of the pairwise event coreference model.

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 event or start the event mention as a new event.

2.1 Algorithm

Formally, let $\{em_i : 1 \le i \le N\}$ be N event mentions in a document and the index i indicates the order it occurs in the document.

Let e_j be the jth event and $f : i \to j$ be the map from event mention index i to event index j.

For each event mention index $k (1 \le k \le N)$, let $I_k = \{t : t = f(i)\ for\ 1 \le i \le k-1\}$ be the set of indices of partially-established events and $E_k = \{e_t : t \in I_k\}$ be the set of partially-established events before the event mention em_k (note that $E_1 = \emptyset$ and $E_2 = \{[em_1]\}$).

We start the iteration from $k = 2$. At each iteration, find the event $e_j \in E_k$ (j is the event index in E_k) such that

$$e_j = \underset{e_t \in E_k}{argmax}(coref(e_t, em_k))$$

where $coref(\cdot,\cdot)$ is called pairwise (event-mention pair) coreference function that computes the coreference score between a prior event and the active event mention.

If the highest score $coref(e_j, em_k)$ is above a threshold δ, we merge em_k into event e_j, otherwise, we start a new event and add it to E_k.

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

We illustrate the agglomerative clustering algorithm for event coreference resolution in Figure 1.

```
Input: event mentions $\{em_i : 1 \le i \le N\}$, coreference
        threshold $\delta$
Output: resolved events $E_{N+1}$
Initialize $E_1 = \emptyset, E_2 = \{[em_1]\}$
for $k = 2$ to $N${
    $j = -1; prob = 0;$
    foreach event $e_t \in E_k$ {
        if ($coref(e_t, em_k) > prob$){
            $j = t; prob = coref(e_t, em_k);$
        }
    }
    if ($prob > \delta$) {
        Extend $e_j$ to $e_j'$ by merging $em_k$ into $e_j$;
        $E_{k+1} = (E_k - \{e_j\}) \cup \{e_j'\}$
    }
    else
        $E_{k+1} = E_k \cup \{[em_k]\}$
}
return $E_{N+1}$
```

Figure 1. Agglomerative clustering algorithm for event coreference resolution

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.

2.2 Pairwise Event Coreference Model

A key issue in the algorithm is how to compute the coreference function $coref(\cdot,\cdot)$ which indicates the probability of merging the active event mention into a prior event. We construct a Maximum-entropy model using the features as tabulated in Table 2. Once the model is trained, it can tell us the probability of merging or not merging for each pair of the active event mention and a prior event. We categorize our features into *baseline*, *distance*, *arguments* and *attributes* feature sets. The four feature sets capture trigger relatedness, trigger distance, argument compatibility and event attribute compatibility respectively.

Table 2. Feature categories for the pairwise event coreference model

Category	**Features**	**Remarks** (AEM: the active event mention, LEM: the last event mention in a prior event)
Baseline	type_subtype	pair of event type and subtype in AEM
	trigger_pair	trigger pair of AEM and LEM
	pos_pair	part-of-speech pair of triggers of AEM and LEM
	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 in AEM matches a trigger spelling in one of the event mentions of the prior event
	stem_match	1 if the trigger stem in AEM and LEM matches a trigger stem in one of the event mentions of the prior event
	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 the prior event
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 the prior event
	prior_num, prior_roles	the number of arguments that only appear in the prior event 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 the prior event
	time_conflict	1 if both AEM and the prior event have an argument with role "Time-Within" and their values conflict
	place_conflict	1 if both AEM and the prior event 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 the prior event conflict

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 an individual component for each of the four event attributes. Such efforts have been largely neglected in the prior research. 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 will study their impact on the system performance. In the next section, we describe the extraction of the four event attributes in details.

3. Extraction of 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 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.

They agree not to {pursue} further cases in foreign courts.

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

I don't think he was {poisoned}.

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

Bush indefinitely postponed a {visit} to Canada.

He failed to be {elected} in the poll.

They abandoned the decision to {purchase} the company.

Some outlets have stopped {selling} lasers

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

We construct a Maximum-entropy model using the following features for this attribute:

- the trigger and its part-of-speech
- 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
- 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.

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. As specified in [1], 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.

We construct a Maximum-entropy model using the following features for this attribute:

- 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
- the first verb within the clause containing the trigger and its POS tag
- 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.

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.

Following lists some GENERIC examples:

Hamas vowed to continue its {attacks}.

Use of the {death} penalty is rare in Indonesia.

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

We observe from the corpus that the GENERIC events are less likely to have "PLACE" argument or "TIME-WITHIN" argument and they tend to have fewer arguments than SPECIFIC events. By taking into account those observations, we construct a Maximum-entropy model using the following features for this attribute:

- 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
- the first verb within the clause containing the trigger and its POS tag
- 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”

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. Following are four examples for PAST event, FUTURE event, PRESENT event and UNSPECIFIED event respectively.

PAST event:

A small group of protesters were {arrested} after they refused to go home.

FUTURE event:

Jean-Rene Fourtou will {replace} Diller as chairman.

PRESENT event:

North Korea and Washington have no formal relations and are still technically at {war}.

UNSPECIFIED event:

It is very legal, and an acceptable policy or practice in Canada to {pay} stipend for people on boards that are active.

We construct a Maximum-entropy model using the following features for this attribute:

- 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
- 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 text types: 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 five-fold cross validation and measured significance with the Wilcoxon signed rank test.

4.2 Performance of 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

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 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). Comparing the third row (perfect, model) with the second row (perfect, majority), we achieved a performance gain of 0.2%, 3.6%,1.8%, 13.4% in Precision (P) for the four attributes Polarity, Modality, Genericity and Tense respectively. Again, when we compare the fifth row (system, model) with the fourth row (system, majority), we achieved a performance gain of 0.5%, 2.6%, 0.7%, 12.7% in Precision (P) respectively, a performance gain of 0.1%, 0.8%, 0.2%, 5.2% in Recall (R) respectively and a performance gain of 0.2%, 1.5%, 0.4%, 8.1% in F-Measure (F) respectively.

It is clear that the improvement for Polarity over the baseline is quite limited since the baseline already obtains a high score. Furthermore, we do not obtain great improvements for Modality and Genericity which may imply that deep parsing features or semantic features should be incorporated into the model for a performance boost. However, we obtain a significant improvement for Tense, either using ground truth event mentions or using system generated event mentions. The score of 0.644 in Precision (third row) implies that there is still a great deal of room for improvement.

Table 4. Performance of 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

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.

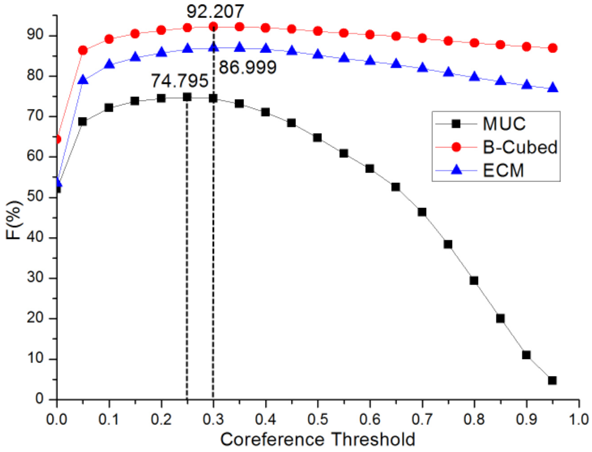

Figure 2. Determining the best coreference threshold δ

Figure 2 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. However, B-Cubed metric consistently reports higher F-scores. As pointed out in [6], B-Cubed metric still suffers from a drawback of giving multiple credits for an individual mention.

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
Baseline	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). Therefore it implies that if we can design better modules to extract event attributes, we will boost our coreference model when practising on system generated event mentions.

We then investigate whether the feature sets have similar impacts on the pairwise event coreference model using the system generated event mentions. The results are presented in Table 6.

Table 6. Feature impact using system generated event mentions

	MUC F	B-Cubed F	ECM F
Baseline	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 *distance* feature set actually hurts the performance, and the *arguments* and *attributes* feature sets only help slightly. The reason may be that all the error propagations from upstream processing before event coreference resolution do have serious impacts on producing the feature values correctly, for example, the trigger labeling may not detect the trigger correctly, thus the distance feature value may not reflect the truth; the argument labeling may not extract the arguments correctly, thus those argument features are not so effective; finally, the event attribute labeling may not label the attributes correctly, thus affect the accuracy of attribute related features.

It implies that if we have better modules for trigger labeling, argument labeling and event attribute labeling, we can produce a higher performance event coreference resolver by exploiting the power of trigger, argument and event attribute related features.

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. To the best of our knowledge, this paper is an early effort to carry out a close study on ACE event coreference resolution and the four event attributes: polarity, modality, genecity and tense.

A variety of methods have been developed for entity coreference resolution (e.g. [9],[10],[11]). The algorithm presented here shares similarity with the pairwise model in [10]. However, in our task, the event mention has much richer structure than the entity mention, thus, it is possible for us to utilize some features that are excluded from entity coreference resolution, e.g., the argument and event attribute related features.

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 the improved components will finally help us improve the performance of event coreference resolution. We also have interests in carrying out research in cross-document event coreference resolution.

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.

7. 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] V. Ng. 2008. Unsupervised models for coreference resolution. *Proc. EMNLP*.

[10] W. M. Soon, H. T. Ng, and D. C. Y. Lim. 2001. A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, 27(4):521–544.

[11] X. Yang, J. Su, J. Lang, C.L. Tan, T. Liu and S. Li. 2008. An Entity-Mention Model for Coreference Resolution with Inductive Logic Programming. *Proc. ACL08*. Columbus, OH.