Refining Event Extraction through Cross-document Inference

Heng Ji **Ralph Grishman**
Computer Science Department
New York University
New York, NY 10003, USA
(hengji, grishman)@cs.nyu.edu

Abstract

We apply the hypothesis of "One Sense Per Discourse" (Yarowsky, 1995) to information extraction (IE), and extend the scope of "discourse" from one single document to a cluster of topically-related documents. We employ a similar approach to propagate consistent event arguments across sentences and documents. Combining global evidence from related documents with local decisions, we design a simple scheme to conduct cross-document inference for improving the ACE event extraction task[1]. Without using any additional labeled data this new approach obtained 7.6% higher F-Measure in trigger labeling and 6% higher F-Measure in argument labeling over a state-of-the-art IE system which extracts events independently for each sentence.

1 Introduction

Identifying events of a particular type within individual documents – 'classical' information extraction – remains a difficult task. Recognizing the different forms in which an event may be expressed, distinguishing events of different types, and finding the arguments of an event are all challenging tasks.

Fortunately, many of these events will be reported multiple times, in different forms, both within the same document and within *topically-related* documents (i.e. a collection of documents sharing participants in potential events). We can take advantage of these alternate descriptions to improve event extraction in the original document, by favoring consistency of interpretation across sentences and documents. Several recent studies involving specific event types have stressed the benefits of going beyond traditional single-document extraction; in particular, Yangarber (2006) has emphasized this potential in his work on medical information extraction. In this paper we demonstrate that appreciable improvements are possible over the variety of event types in the ACE (Automatic Content Extraction) evaluation through the use of cross-sentence and cross-document evidence.

As we shall describe below, we can make use of consistency at several levels: consistency of word sense across different instances of the same word in related documents, and consistency of arguments and roles across different mentions of the same or related events. Such methods allow us to build dynamic background knowledge as required to interpret a document and can compensate for the limited annotated training data which can be provided for each event type.

2 Task and Baseline System

2.1 ACE Event Extraction Task

The event extraction task we are addressing is that of the Automatic Content Extraction (ACE) evaluations[2]. ACE defines the following terminology:

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

[2] In this paper we don't consider event mention coreference resolution and so don't distinguish event mentions and events.

Proceedings of ACL-08: HLT, pages 254–262,
Columbus, Ohio, USA, June 2008.

entity: an object or a set of objects in one of the semantic categories of interest
mention: a reference to an entity (typically, a noun phrase)
event trigger: the main word which most clearly expresses an event occurrence
event arguments: the mentions that are involved in an event (participants)
event mention: a phrase or sentence within which an event is described, including trigger and arguments

The 2005 ACE evaluation had 8 types of events, with 33 subtypes; for the purpose of this paper, we will treat these simply as 33 distinct event types. For example, for a sentence:

Barry Diller on Wednesday quit as chief of Vivendi Universal Entertainment.

the event extractor should detect a "Personnel_End-Position" event mention, with the trigger word, the position, the person who quit the position, the organization, and the time during which the event happened:

Trigger		*Quit*
Arguments	Role = Person	*Barry Diller*
	Role = Organization	*Vivendi Universal Entertainment*
	Role = Position	*Chief*
	Role = Time-within	*Wednesday*

Table 1. Event Extraction Example

We define the following standards to determine the *correctness* of an event mention:

- *A trigger is correctly labeled* if its event type and offsets match a reference trigger.
- *An argument is correctly identified* if its event type and offsets match any of the reference argument mentions.
- *An argument is correctly identified and classified* if its event type, offsets, and role match any of the reference argument mentions.

2.2 A Baseline Within-Sentence Event Tagger

We use a state-of-the-art English IE system as our baseline (Grishman et al., 2005). This system extracts events independently for each sentence. Its training and test procedures are as follows.

The system combines pattern matching with statistical models. For every event mention in the ACE training corpus, patterns are constructed based on the sequences of constituent heads separating the trigger and arguments. In addition, a set of Maximum Entropy based classifiers are trained:

- Trigger Labeling: to distinguish event mentions from non-event-mentions, to classify event mentions by type;
- Argument Classifier: to distinguish arguments from non-arguments;
- Role Classifier: to classify arguments by argument role.
- Reportable-Event Classifier: Given a trigger, an event type, and a set of arguments, to determine whether there is a reportable event mention.

In the test procedure, each document is scanned for instances of triggers from the training corpus. When an instance is found, the system tries to match the environment of the trigger against the set of patterns associated with that trigger. This pattern-matching process, if successful, will assign some of the mentions in the sentence as arguments of a potential event mention. The argument classifier is applied to the remaining mentions in the sentence; for any argument passing that classifier, the role classifier is used to assign a role to it. Finally, once all arguments have been assigned, the reportable-event classifier is applied to the potential event mention; if the result is successful, this event mention is reported.

3 Motivations

In this section we shall present our motivations based on error analysis for the baseline event tagger.

3.1 One Trigger Sense Per Cluster

Across a heterogeneous document corpus, a particular verb can sometimes be trigger and sometimes not, and can represent different event types. However, for a collection of topically-related documents, the distribution may be much more convergent. We investigate this hypothesis by automatically obtaining 25 related documents for each test text. The statistics of some trigger examples are presented in table 2.

Candidate Triggers		Event Type	Perc./Freq. as trigger in ACE training corpora	Perc./Freq. as trigger in test document	Perc./Freq. as trigger in test + related documents
Correct Event Triggers	*advance*	Movement_Transport	31% of 16	50% of 2	88.9% of 27
	fire	Personnel_End-Position	7% of 81	100% of 2	100% of 10
	fire	Conflict_Attack	54% of 81	100% of 3	100% of 19
	replace	Personnel_End-Position	5% of 20	100% of 1	83.3% of 6
	form	Business_Start-Org	12% of 8	100% of 2	100% of 23
	talk	Contact_Meet	59% of 74	100% of 4	100% of 26
Incorrect Event Triggers	*hurt*	Life_Injure	24% of 33	0% of 2	0% of 7
	execution	Life_Die	12% of 8	0% of 4	4% of 24

Table 2. Examples: Percentage of a Word as Event Trigger in Different Data Collections

As we can see from the table, the likelihood of a candidate word being an event trigger in the test document is closer to its distribution in the collection of related documents than the uniform training corpora. So if we can determine the sense (event type) of a word in the related documents, this will allow us to infer its sense in the test document. In this way related documents can help recover event mentions missed by within-sentence extraction.

For example, in a document about "the advance into Baghdad":

Example 1:
[***Test Sentence***]
Most US army commanders believe it is critical to pause the breakneck ***advance*** *towards Baghdad to secure the supply lines and make sure weapons are operable and troops resupplied….*
[***Sentences from Related Documents***]
British and US forces report gains in the ***advance*** *on Baghdad and take control of Umm Qasr, despite a fierce sandstorm which slows another flank.*
…

The baseline event tagger is not able to detect "advance" as a "Movement_Transport" event trigger because there is no pattern "advance towards [Place]" in the ACE training corpora ("advance" by itself is too ambiguous). The training data, however, does include the pattern "advance on [Place]", which allows the instance of "advance" in the related documents to be successfully identified with high confidence by pattern matching as an event. This provides us much stronger "feedback" confidence in tagging 'advance' in the test sentence as a correct trigger.

On the other hand, if a word is not tagged as an event trigger in most related documents, then it's less likely to be correct in the test sentence despite its high local confidence. For example, in a document about "assessment of Russian president Putin":

Example 2:
[***Test Sentence***]
But few at the Kremlin forum suggested that Putin's own standing among voters will be ***hurt*** *by Russia's apparent diplomacy failures.*
[***Sentences from Related Documents***]
Putin boosted ties with the United States by throwing his support behind its war on terrorism after the Sept. 11 attacks, but the Iraq war has ***hurt*** *the relationship.*
…

The word "hurt" in the test sentence is mistakenly identified as a "Life_Injure" trigger with high local confidence (because the within-sentence extractor misanalyzes "voters" as the object of "hurt" and so matches the pattern "[Person] be hurt"). Based on the fact that many other instances of "hurt" are not "Life_Injure" triggers in the related documents, we can successfully remove this wrong event mention in the test document.

3.2 One Argument Role Per Cluster

Inspired by the observation about trigger distribution, we propose a similar hypothesis – one argument role per cluster for event arguments. In other words, each entity plays the same argument role, or no role, for events with the same type in a collection of related documents. For example,

Example 3:
[***Test Sentence***]
Vivendi *earlier this week confirmed months of press speculation that it planned to* ***shed*** *its entertainment assets by the end of the year.*
[***Sentences from Related Documents***]
Vivendi *has been trying to* ***sell*** *assets to pay off huge debt, estimated at the end of last month at more than $13 billion.*
Under the reported plans, Blackstone Group would ***buy Vivendi****'s theme park division, including Universal Studios Hollywood, Universal Orlando in Florida...*
…

The above test sentence doesn't include an explicit trigger word to indicate "Vivendi" as a "seller" of a "Transaction_Transfer-Ownership" event mention, but "Vivendi" is correctly identified as "seller" in many other related sentences (by matching patterns "[Seller] sell" and "buy [Seller]'s"). So we can incorporate such additional information to enhance the confidence of "Vivendi" as a "seller" in the test sentence.

On the other hand, we can remove spurious arguments with low cross-document frequency and confidence. In the following example,

Example 4:
[***Test Sentence***]
The Davao Medical Center*, a regional government hospital, recorded 19 deaths with 50 wounded.*

"the Davao Medical Center" is mistakenly tagged as "Place" for a "Life_Die" event mention. But the same annotation for this mention doesn't appear again in the related documents, so we can determine it's a spurious argument.

4 System Approach Overview

Based on the above motivations we propose to incorporate global evidence from a cluster of related documents to refine local decisions. This section gives more details about the baseline within-sentence event tagger, and the information retrieval system we use to obtain related documents. In the next section we shall focus on describing the inference procedure.

4.1 System Pipeline

Figure 1 depicts the general procedure of our approach. *EMSet* represents a set of event mentions which is gradually updated.

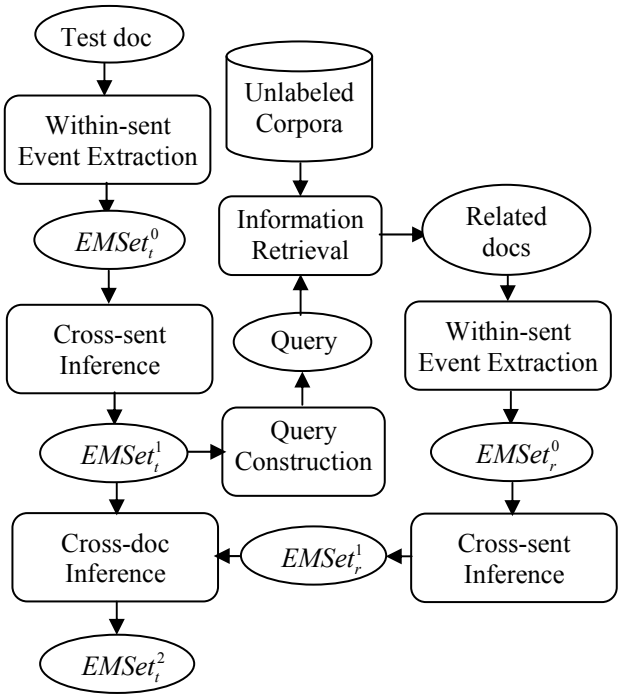

Figure 1. Cross-doc Inference for Event Extraction

4.2 Within-Sentence Event Extraction

For each event mention in a test document t, the baseline Maximum Entropy based classifiers produce three types of confidence values:

- *LConf(trigger,etype)*: The probability of a string *trigger* indicating an event mention with type *etype*; if the event mention is produced by pattern matching then assign confidence 1.
- *LConf(arg, etype)*: The probability that a mention *arg* is an argument of some particular event type *etype*.
- *LConf(arg, etype, role)*: If *arg* is an argument with event type *etype*, the probability of *arg* having some particular *role*.

We apply within-sentence event extraction to get an initial set of event mentions $EMSet_t^0$, and conduct cross-sentence inference (details will be presented in section 5) to get an updated set of event mentions $EMSet_t^1$.

4.3 Information Retrieval

We then use the INDRI retrieval system (Strohman et al., 2005) to obtain the top N (N=25 in this pa-

per[3]) related documents. We construct an INDRI query from the triggers and arguments, each weighted by local confidence and frequency in the test document. For each argument we also add other names coreferential with or bearing some ACE relation to the argument.

For each related document r returned by INDRI, we repeat the within-sentence event extraction and cross-sentence inference procedure, and get an expanded event mention set $EMSet^1_{t+r}$. Then we apply cross-document inference to $EMSet^1_{t+r}$ and get the final event mention output $EMSet^2_t$.

5 Global Inference

The central idea of inference is to obtain document-wide and cluster-wide statistics about the frequency with which triggers and arguments are associated with particular types of events, and then use this information to correct event and argument identification and classification.

For a set of event mentions we tabulate the following document-wide and cluster-wide confidence-weighted frequencies:

- for each trigger string, the frequency with which it appears as the trigger of an event of a particular type;
- for each event argument string and the names coreferential with or related to the argument, the frequency of the event type;
- for each event argument string and the names coreferential with or related to the argument, the frequency of the event type and role.

Besides these frequencies, we also define the following *margin* metric to compute the confidence of the best (most frequent) event type or role:

Margin =
(WeightedFrequency (most frequent value)
– WeightedFrequency (second most freq value))/
WeightedFrequency (second most freq value)

A large margin indicates greater confidence in the most frequent value. We summarize the frequency and confidence metrics in Table 3.

Based on these confidence metrics, we designed the inference rules in Table 4. These rules are applied in the order (1) to (9) based on the principle of improving 'local' information before global propagation. Although the rules may seem complex, they basically serve two functions:

- to remove triggers and arguments with low (local or cluster-wide) confidence;
- to adjust trigger and argument identification and classification to achieve (document-wide or cluster-wide) consistency.

6 Experimental Results and Analysis

In this section we present the results of applying this inference method to improve ACE event extraction.

6.1 Data

We used 10 newswire texts from ACE 2005 training corpora (from March to May of 2003) as our development set, and then conduct blind test on a separate set of 40 ACE 2005 newswire texts. For each test text we retrieved 25 related texts from English TDT5 corpus which in total consists of 278,108 texts (from April to September of 2003).

6.2 Confidence Metric Thresholding

We select the thresholds (δ_k with k=1~13) for various confidence metrics by optimizing the F-measure score of each rule on the development set, as shown in Figure 2 and 3 as follows.

Each curve in Figure 2 and 3 shows the effect on precision and recall of varying the threshold for an individual rule.

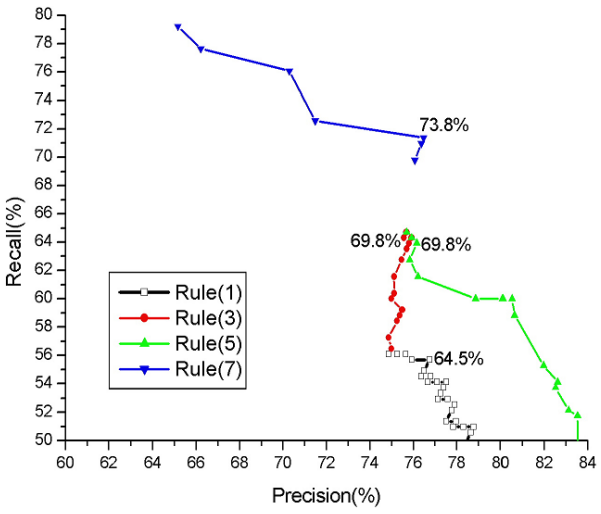

Figure 2. Trigger Labeling Performance with Confidence Thresholding on Dev Set

[3] We tested different N ∈ [10, 75] on dev set; and N=25 achieved best gains.

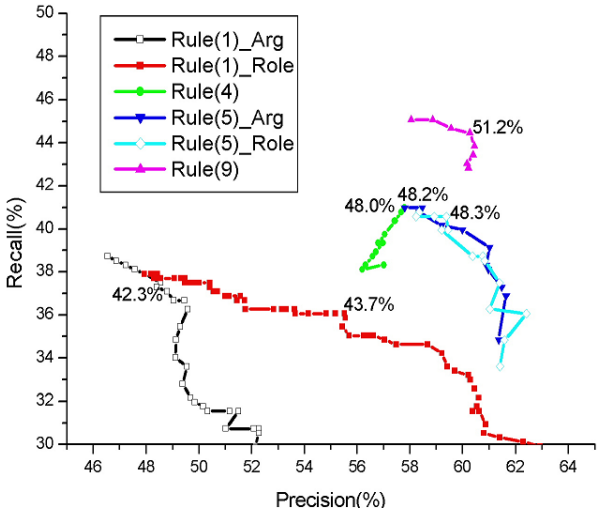

Figure 3. Argument Labeling Performance with Confidence Thresholding on Dev Set

The labeled point on each curve shows the best F-measure that can be obtained on the development set by adjusting the threshold for that rule. The gain obtained by applying successive rules can be seen in the progression of successive points towards higher recall and, for argument labeling, precision[4].

6.3 Overall Performance

Table 5 shows the overall Precision (P), Recall (R) and F-Measure (F) scores for the blind test set. In addition, we also measured the performance of two human annotators who prepared the ACE 2005 training data on 28 newswire texts (a subset of the blind test set). The final key was produced by review and adjudication of the two annotations.

Both cross-sentence and cross-document inferences provided significant improvement over the baseline with local confidence thresholds controlled.

We conducted the Wilcoxon Matched-Pairs Signed-Ranks Test on a document basis. The results show that the improvement using cross-sentence inference is significant at a 99.9% confidence level for both trigger and argument labeling; adding cross-document inference is significant at a 99.9% confidence level for trigger labeling and 93.4% confidence level for argument labeling.

[4] We didn't show the classification adjusting rules (2), (6) and (8) here because of their relatively small impact on dev set.

6.4 Discussion

From table 5 we can see that for trigger labeling our approach dramatically enhanced recall (22.9% improvement) with some loss (7.4%) in precision. This precision loss was much larger than that for the development set (0.3%). This indicates that the trigger propagation thresholds optimized on the development set were too low for the blind test set and thus more spurious triggers got propagated. The improved trigger labeling is better than one human annotator and only 4.7% worse than another.

For argument labeling we can see that cross-sentence inference improved both identification (3.7% higher F-Measure) and classification (6.1% higher accuracy); and cross-document inference mainly provided further gains (1.9%) in classification. This shows that identification consistency may be achieved within a narrower context while the classification task favors more global background knowledge in order to solve some difficult cases. This matches the situation of human annotation as well: we may decide whether a mention is involved in some particular event or not by reading and analyzing the target sentence itself; but in order to decide the argument's role we may need to frequently refer to wider discourse in order to infer and confirm our decision. In fact sometimes it requires us to check more similar web pages or even wikipedia databases. This was exactly the intuition of our approach. We should also note that human annotators label arguments based on perfect entity mentions, but our system used the output from the IE system. So the gap was also partially due to worse entity detection.

Error analysis on the inference procedure shows that the propagation rules (3), (4), (7) and (9) produced a few extra false alarms. For trigger labeling, most of these errors appear for support verbs such as "take" and "get" which can only represent an event mention together with other verbs or nouns. Some other errors happen on nouns and adjectives. These are difficult tasks even for human annotators. As shown in table 5 the inter-annotator agreement on trigger identification is only about 40%. Besides some obvious overlooked cases (it's probably difficult for a human to remember 33 different event types during annotation), most difficulties were caused by judging generic verbs, nouns and adjectives.

Performance / System/Human	Trigger Identification +Classification			Argument Identification			Argument Classification Accuracy	Argument Identification +Classification		
	P	R	F	P	R	F		P	R	F
Within-Sentence IE with Rule (1) (Baseline)	67.6	53.5	59.7	47.8	38.3	42.5	86.0	41.2	32.9	36.6
Cross-sentence Inference	64.3	59.4	61.8	54.6	38.5	45.1	90.2	49.2	34.7	40.7
Cross-sentence+ Cross-doc Inference	**60.2**	**76.4**	**67.3**	**55.7**	**39.5**	**46.2**	**92.1**	**51.3**	**36.4**	**42.6**
Human Annotator1	59.2	59.4	59.3	60.0	69.4	64.4	85.8	51.6	59.5	55.3
Human Annotator2	69.2	75.0	72.0	62.7	85.4	72.3	86.3	54.1	73.7	62.4
Inter-Annotator Agreement	41.9	38.8	40.3	55.2	46.7	50.6	91.7	50.6	42.9	46.4

Table 5. Overall Performance on Blind Test Set (%)

In fact, compared to a statistical tagger trained on the corpus after expert adjudication, a human annotator tends to make more mistakes in trigger classification. For example it's hard to decide whether "named" represents a "Personnel_Nominate" or "Personnel_Start-Position" event mention; "hacked to death" represents a "Life_Die" or "Conflict_Attack" event mention without following more specific annotation guidelines.

7 Related Work

The trigger labeling task described in this paper is in part a task of word sense disambiguation (WSD), so we have used the idea of sense consistency introduced in (Yarowsky, 1995), extending it to operate across related documents.

Almost all the current event extraction systems focus on processing single documents and, except for coreference resolution, operate a sentence at a time (Grishman et al., 2005; Ahn, 2006; Hardy et al., 2006).

We share the view of using global inference to improve event extraction with some recent research. Yangarber et al. (Yangarber and Jokipii, 2005; Yangarber, 2006; Yangarber et al., 2007) applied cross-document inference to correct local extraction results for disease name, location and start/end time. Mann (2007) encoded specific inference rules to improve extraction of CEO (name, start year, end year) in the MUC management succession task. In addition, Patwardhan and Riloff (2007) also demonstrated that selectively applying event patterns to relevant regions can improve MUC event extraction. We expand the idea to more general event types and use information retrieval techniques to obtain wider background knowledge from related documents.

8 Conclusion and Future Work

One of the initial goals for IE was to create a database of relations and events from the entire input corpus, and allow further logical reasoning on the database. The artificial constraint that extraction should be done independently for each document was introduced in part to simplify the task and its evaluation. In this paper we propose a new approach to break down the document boundaries for event extraction. We gather together event extraction results from a set of related documents, and then apply inference and constraints to enhance IE performance.

In the short term, the approach provides a platform for many byproducts. For example, we can naturally get an event-driven summary for the collection of related documents; the sentences including high-confidence events can be used as additional training data to bootstrap the event tagger; from related events in different timeframes we can derive entailment rules; the refined consistent events can serve better for other NLP tasks such as template based question-answering. The aggregation approach described here can be easily extended to improve relation detection and coreference resolution (two argument mentions referring to the same role of related events are likely to corefer). Ultimately we would like to extend the system to perform essential, although probably lightweight, event prediction.

XSent-Trigger-Freq(trigger, etype)	The weighted frequency of string *trigger* appearing as the trigger of an event of type *etype* across all sentences within a document
XDoc-Trigger-Freq (trigger, etype)	The weighted frequency of string *trigger* appearing as the trigger of an event of type *etype* across all documents in a cluster
XDoc-Trigger-BestFreq (trigger)	Maximum over all *etype*s of *XDoc-Trigger-Freq (trigger, etype)*
XDoc-Arg-Freq(arg, etype)	The weighted frequency of *arg* appearing as an argument of an event of type *etype* across all documents in a cluster
XDoc-Role-Freq(arg, etype, role)	The weighted frequency of *arg* appearing as an argument of an event of type *etype* with role *role* across all documents in a cluster
XDoc-Role-BestFreq(arg)	Maximum over all etypes and roles of *XDoc-Role-Freq(arg, etype, role)*
XSent-Trigger-Margin(trigger)	The margin value of *trigger* in *XSent-Trigger-Freq*
XDoc-Trigger-Margin(trigger)	The margin value of *trigger* in *XDoc-Trigger-Freq*
XDoc-Role-Margin(arg)	The margin value of *arg* in *XDoc-Role-Freq*

Table 3. Global Frequency and Confidence Metrics

Rule (1): Remove Triggers and Arguments with Low Local Confidence
If *LConf(trigger, etype)* < δ_1, then delete the whole event mention *EM*; If *LConf(arg, etype)* < δ_2 or *LConf(arg, etype, role)* < δ_3, then delete *arg*.
Rule (2): Adjust Trigger Classification to Achieve Document-wide Consistency
If *XSent-Trigger-Margin(trigger)* > δ_4, then propagate the most frequent *etype* to all event mentions with *trigger* in the document; and correct roles for corresponding arguments.
Rule (3): Adjust Trigger Identification to Achieve Document-wide Consistency
If *LConf(trigger, etype)* > δ_5, then propagate *etype* to all unlabeled strings *trigger* in the document.
Rule (4): Adjust Argument Identification to Achieve Document-wide Consistency
If *LConf(arg, etype)* > δ_6, then in the document, for each sentence containing an event mention *EM* with *etype*, add any unlabeled mention in that sentence with the same head as *arg* as an argument of *EM* with *role*.
Rule (5): Remove Triggers and Arguments with Low Cluster-wide Confidence
If *XDoc-Trigger-Freq (trigger, etype)* < δ_7, then delete *EM*; If *XDoc-Arg-Freq(arg, etype)* < δ_8 or *XDoc-Role-Freq(arg, etype, role)* < δ_9, then delete *arg*.
Rule (6): Adjust Trigger Classification to Achieve Cluster-wide Consistency
If *XDoc-Trigger-Margin(trigger)* > δ_{10}, then propagate most frequent *etype* to all event mentions with *trigger* in the cluster; and correct roles for corresponding arguments.
Rule (7): Adjust Trigger Identification to Achieve Cluster-wide Consistency
If *XDoc-Trigger-BestFreq (trigger)* > δ_{11}, then propagate *etype* to all unlabeled strings *trigger* in the cluster, override the results of Rule (3) if conflict.
Rule (8): Adjust Argument Classification to Achieve Cluster-wide Consistency
If *XDoc-Role-Margin(arg)* > δ_{12}, then propagate the most frequent *etype* and *role* to all arguments with the same head as *arg* in the entire cluster.
Rule (9): Adjust Argument Identification to Achieve Cluster-wide Consistency
If *XDoc-Role-BestFreq(arg)* > δ_{13}, then in the cluster, for each sentence containing an event mention *EM* with *etype*, add any unlabeled mention in that sentence with the same head as *arg* as an argument of *EM* with *role*.

Table 4. Probabilistic Inference Rule

Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency under Contract No. HR0011-06-C-0023, and the National Science Foundation under Grant IIS-00325657. 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.

References

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

Ralph Grishman, David Westbrook and Adam Meyers. 2005. NYU's English ACE 2005 System Description. *Proc. ACE 2005 Evaluation Workshop*. Washington, US.

Hilda Hardy, Vika Kanchakouskaya and Tomek Strzalkowski. 2006. Automatic Event Classification Using Surface Text Features. *Proc. AAAI06 Workshop on Event Extraction and Synthesis*. Boston, Massachusetts. US.

Gideon Mann. 2007. Multi-document Relationship Fusion via Constraints on Probabilistic Databases. *Proc. HLT/NAACL 2007*. Rochester, NY, US.

Siddharth Patwardhan and Ellen Riloff. 2007. Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions. *Proc. EMNLP 2007.* Prague, Czech Republic.

Trevor Strohman, Donald Metzler, Howard Turtle and W. Bruce Croft. 2005. Indri: A Language-model based Search Engine for Complex Queries (extended version). *Technical Report IR-407, CIIR,* Umass Amherst, US.

Roman Yangarber, Clive Best, Peter von Etter, Flavio Fuart, David Horby and Ralf Steinberger. 2007. Combining Information about Epidemic Threats from Multiple Sources. *Proc. RANLP 2007 workshop on Multi-source, Multilingual Information Extraction and Summarization*. Borovets, Bulgaria.

Roman Yangarber. 2006. Verification of Facts across Document Boundaries. *Proc. International Workshop on Intelligent Information Access*. Helsinki, Finland.

Roman Yangarber and Lauri Jokipii. 2005. Redundancy-based Correction of Automatically Extracted Facts. *Proc. HLT/EMNLP 2005*. Vancouver, Canada.

David Yarowsky. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *Proc. ACL 1995*. Cambridge, MA, US.