

# Acquiring Topic Features to Improve Event Extraction: in Pre-selected and Balanced Collections

**Shasha Liao**

Computer Science Department  
New York University  
liaoss@cs.nyu.edu

**Ralph Grishman**

Computer Science Department  
New York University  
grishman@cs.nyu.edu

## Abstract

Event extraction is a particularly challenging type of information extraction (IE) that may require inferences from the whole article. However, most current event extraction systems rely on local information at the phrase or sentence level, and do not consider the article as a whole, thus limiting extraction performance. Moreover, most annotated corpora are artificially enriched to include enough positive samples of the events of interest; event identification on a more balanced collection, such as unfiltered newswire, may perform much worse. In this paper, we investigate the use of unsupervised topic models to extract topic features to improve event extraction both on test data similar to training data, and on more balanced collections. We compare this unsupervised approach to a supervised multi-label text classifier, and show that unsupervised topic modeling can get better results for both collections, and especially for a more balanced collection. We show that the unsupervised topic model can improve trigger, argument and role labeling by 3.5%, 6.9% and 6% respectively on a pre-selected corpus, and by 16.8%, 12.5% and 12.7% on a balanced corpus.

## 1 Introduction

The goal of event extraction is to identify instances of a class of events in free text, along with their arguments. In this paper, we focus on the ACE 2005 event extraction task, which

involved a set of 33 generic event types and subtypes appearing frequently in the news. It generally expresses the core arguments plus place and time information of a single event, like *Attack*, *Marry* or *Arrest*.

In general, identifying an ACE event can be quite difficult. Given a narrow scope of information, even a human cannot make a confident decision. For example, for the sentence:

(1) *So he returned to combat ...*

it is hard to tell whether it is an *Attack* event, which is defined as a violent physical act causing harm or damage, or whether it refers to a more innocent endeavor such as a tennis match. A broader field of view is often helpful to understand how facts tie together. If we read the whole article, and find it to be a terrorist story, it is easy to tag this as an *Attack* event; however, if it is in a tennis report, we probably won't tag it as an *Attack* event.

The problem of event identification is exacerbated if we shift to corpora with a topic distribution different from the training and official test corpus. In general, an effort is made to have the test corpora be representative of the sort of texts to which the NLP process is intended to be applied. In the case of the event extraction, this has generally been news sources such as newswires or broadcast news transcripts. However, a particular event type is likely to occur infrequently in the general news, which might contain many different topics, only a few of which are likely to include mentions of this event type. As a result, a typical evaluation corpus (a few hundred hand-annotated documents), if selected at random, would contain only a few events, which is not sufficient for training. To avoid this, these annotated corpora are artificially enriched through a combination of topic classification and manual review, so that

they contain a high concentration of the events of interest. For example, in the MUC-3/4 test corpora, about 60% of the documents include relevant events, and in the ACE 2005 training corpus 48% include *Attack* events.

If we train and test the event extraction system on ACE annotated corpora, the problem epitomized by (1) is not significant because there are very few sports articles in the ACE evaluation: 74% of the instances of the word “combat” indicate an *Attack* event. However, if you extend the evaluation to a more balanced collection, for example, the un-filtered New York Times (NYT) newswire, you will find that there are a lot of sports articles and an event extractor will mistakenly tag lots of sports events as *Attack* events. Grishman (2010) drew attention to this phenomenon, pointing out that only about 17% of articles from the contemporaneous sample of The NYT newswire contained attack events, compared to 48% in the ACE evaluation. In this situation, if we apply the event extractor trained on the ACE corpus to the balanced NYT newswire, the performance may be significantly degraded.

Clearly, the topic of the document is a good predictor of particular event types. For example, a reference to “war” inside a business article might refer to a financial competition; while “war” inside a military article would be more likely to refer to a physical attack event. Text classification is used here to identify document topic, and the final decision can be made based on both local evidence and document relevance (Grishman 2010). However, this method has three disadvantages:

First, the event type and document topic are not always strongly connected, and it depends significantly on what kind of event we are going to explore. If the events are related to the main category of the article, only knowing the article category is enough. But if they are not, treating each document as a single topic is not enough. For example, *Die* events might appear in military, financial, political or even sports articles. And most of the time, it is not the main event reported by the article. The article may focus more on the reason for the death, the biography of the person, or the effect of the death.

Second, when the article talks about more than one scenario, simple text classification will basically ignore the secondary scenario. For example, if a sports article that reported the results of a football game also mentions a fight between the fans of two teams, the topic of the

document might be “sports”, which is irrelevant to *Attack* events; however, there is an *Attack* event, which appears in the secondary scenario of the document.

Third, the category or relevance depends on the annotated data, and a classifier may be unable to deal with articles whose topics were rarely seen in the training data. Thus, if the category distribution of the evaluation data is different from the training data, a text classifier might have poor performance.

To solve the first two problems, we need to treat each document as a mixture of several topics instead of one; to solve the third problem, we want to see if unsupervised methods can give us some guidance which a supervised method cannot. These two goals are easily connected to a topic model, for example, Latent Dirichlet Allocation.

## 2 ACE Event Extraction

In this section, we will describe the ACE event extraction task and explain why it is difficult.

### 2.1 Task Description

ACE defines an event as a specific occurrence involving participants<sup>1</sup>, and it annotates 8 types and 33 subtypes of events. In this task, an *event mention* is a phrase or sentence within which an event is described, including trigger and arguments. An event mention must have one and only one trigger, and can have an arbitrary number of arguments. The *event trigger* is the main word that most clearly expresses an event occurrence. The *event mention arguments (roles)*<sup>2</sup> are the entity mentions that are involved in an event mention, and their relation to the event. For example, an event “attack” might include participants like “attacker” or “target”, or attributes like “time within” and “place”. Arguments will be taggable only when they occur within the scope of the corresponding event, typically the same sentence.

Here is an example:

(4) *Three murders occurred in France today, including the senseless slaying of Bob*

---

<sup>1</sup> See [http://projects.ldc.upenn.edu/ace/docs/English-Events-Guidelines\\_v5.4.3.pdf](http://projects.ldc.upenn.edu/ace/docs/English-Events-Guidelines_v5.4.3.pdf) for a description of this task.

<sup>2</sup> Note that we do not deal with event mention coreference in this paper, so each event mention is treated as a separate event.

*Cole. Bob was on his way home when he was attacked...*

There are two *Die* events, which share the same *Place* and *Time* roles, with different *Victim* roles. And there is one *Attack* event sharing the same *Place* and *Time* roles with the *Die* events.

Event type	Trigger	Role		
		Place	Victim	Time
Die	murder	France		today
Die	slaying	France	Bob Cole	today
Event type	Trigger	Role		
		Place	Target	Time
Attack	attack	France	Bob	today

Table 1. An example of event trigger and roles

## 2.2 Problems

Identifying the trigger – the word most clearly expressing the event - is essential for event extraction. Usually, the trigger itself is the most important clue in detecting and classifying the type of an event. For example, the word “attack” is very likely to represent an *Attack* event while the word “meet” is not. However, this is not always enough. If we collect all the words that serve as an event trigger at least once, and plot their probability of triggering an event (Figure 1), we see that the probabilities are widely scattered. Some words always trigger an event (probability = 1.0), but most are ambiguous.

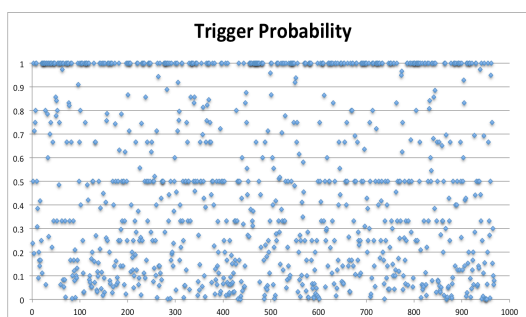


Figure 1. Distribution of trigger probability (X axis represents the words in alphabetical order)

Why is identifying an event so difficult? First of all, a word may be ambiguous and have several senses, only some of which correspond to a particular event type. Moreover, identifying the correct sense is not enough: several different senses of a word might refer to the same event type, and the same sense does not guarantee the occurrence of the specific event: the arguments need to be considered as well. Take the word

“shoot”, for example; the senses “hit with a missile from a weapon” and “fire a shot” might both predicate an *Attack* event, but to guarantee that, we need to not only identify its sense is, for example, “fire a shot”, not “record on photographic film”, but also identify that its target is a person, organization, Geo-Political Entity (GPE), weapon or facility, not an animal. Hunting-related or shooting-contest-related activities should not be tagged as *Attack* events.

Thus, the identification of the trigger and the arguments interact: the relation between the trigger and the argument is one essential factor to identify both the trigger and the role of the argument. For example, if we know that the object of the word “shoot” is a person and it has the “fire a shot” sense, we can confidently identify the person as the *Target* role, and tag “shoot” as the trigger of an *Attack* event.

As a result, most current event extraction systems consider trigger and argument information together to tag a reportable event (see the baseline system in section 5.1).

## 3 Related Work

To the best of our knowledge, we are the first to use unsupervised topic models in event extraction. However, there are some similar approaches that consider the relevance of the document to the specific scenario or event type. For scenario extraction in MUC-3/4, Riloff (1996) initiated this approach and claimed that if a corpus can be divided into documents involving a certain event type and those not involving that type, patterns can be evaluated based on their frequency in relevant and irrelevant documents. Yangarber et al. (2000) incorporated Riloff’s metric into a bootstrapping procedure. Patwardhan and Riloff (2007) presented an information extraction system that finds relevant regions of text and applies extraction patterns within those regions. Liao and Grishman (2010b) also pointed out that the pre-selection of the bootstrapping corpus (based on document topic) is quite essential to this approach. Although their approach involved bootstrapping, it gives the intuition that the event/scenario and the document topic are strongly connected.

For ACE event extraction, most current systems focus on processing one sentence at a time (Grishman et al., 2005; Ahn, 2006; Hardy et al. 2006). However, there have been several studies using high-level information at the document level. Finkel et al. (2005) used Gibbs

sampling, a simple Monte Carlo method used to perform approximate inference in factored probabilistic models. By using simulated annealing in place of Viterbi decoding in sequence models such as HMMs, CMMs, and CRFs, it is possible to incorporate non-local structure while preserving tractable inference. They used this technique to augment an information extraction system with long-distance dependency models, enforcing label consistency and extraction template consistency constraints. Ji and Grishman (2008) extended the scope from a single document to a cluster of topic-related documents and employed a rule-based approach to propagate consistent trigger classification and event arguments across sentences and documents. Liao and Grishman (2010a) extended this consistency within each event type to a distribution among different event types, and obtained an appreciable improvement in both event and event argument identification.

There is not as much work on evaluation on a more balanced collection when the training corpus has a different distribution. Grishman (2010) first pointed out that understanding the characteristics of the corpus is an inherent part of the event extraction task. He gave a small example of the effect of applying an event extractor to a more balanced corpus, and used a document classifier to reduce the spurious errors.

#### 4 Topic Features in Event Extraction

Most previous studies that acquire wider scope information use preselected corpora, like (Riloff 1996); or are rule-based, like Ji and Grishman (2008); or involve supervised learning from the same training data, like Finkel et al. (2005), Liao and Grishman (2010a). We are more interested in using a topic model to provide such information.

A topic model, like Latent Dirichlet Allocation (LDA), is a generative model that allows sets of observations to be explained by unobserved groups. For example, if the observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word is attributable to one of the document's topics. For event extraction, there is a similar assumption that each document consists of various events, and each event is presented by one or several snippets in the document. We want to know if these two can be somehow connected and how one can improve the other.

In this paper, we are more interested in an unsupervised approach from a large untagged corpus. In this way, we can avoid the data bias that may be introduced by an unrepresentative training collection, thus providing better high-level information than previous approaches, especially when applied to the final target application instead of a specially selected development or evaluation corpus.

##### 4.1 Features from Unsupervised Topic Model (LDA)

Latent Dirichlet Allocation (LDA) tries to group words into “topics”, where each word is generated from a single topic, and different words in a document may be generated from different topics. Thus, each document is represented as a list of mixing proportions for these mixture components and thereby reduced to a probability distribution on a fixed set of topics. In LDA, each document may be viewed as a mixture of various topics. A document is generated by picking a distribution over topics, and given this distribution, picking the topic of each specific word to be generated. Then words are generated given their topics. Words are considered to be independent given the topics; this is a standard bag of words model assumption where individual words are exchangeable.

Unlike supervised classification, there are no explicit labels, like “finance” or “war”, in unsupervised LDA. Instead, we can imagine each topic as “a cluster of words that refers to an implicit topic”. For example, if a document contains words like “company”, “financial”, and “market”, we assume it contains a “financial topic” and are more confident to find events like *Start-Position*, *End-Position*, while a document that contains “war”, “combat”, “fire”, and “force” will be assumed to contain the “war topic”, which is more likely to contain *Attack*, *Die*, or *Injure* events.

##### 4.2 Features from Multi-label Text Classifier

As the event extraction system uses a supervised model, it is natural to ask whether supervised topic features are better than unsupervised ones. There are several possible approaches. For example, we can first run a topic classification filter to predict whether or not a document is likely to contain a specific type of event. However, because of the limited precision of a simple classifier such as a bag-of-words MaxEnt classifier (for *Attack* events, the precision is

around 69% in ACE data), using it as a pre-filter will lead to event recall or precision errors. Instead, we decide to use the topic information as features within the event extraction system. As one document might contain several event types, we tag each document with labels indicating the presence of one or more events of a given type, which is a multi-label text classification problem. In this section, we build a supervised multi-label text classifier to compare to the unsupervised topic model.

The basic idea for a multi-label classifier comes from the *credit attribution* problem in social bookmarking websites, where pages have multiple tags, but the tags do not always apply with equal specificity across the whole page (Ramage et al. 2009). This relation between tag and page is quite similar to that between event and document, because one document might also have multiple events of differing specificity. For example, an *Attack* event may be more related to the main topic of the document than a *Meet* event.

We use Labeled LDA (L-LDA) to build the multi-label text classifier, which is reported (Ramage et al. 2009) to outperform SVMs when extracting tag-specific document snippets, and is competitive with SVMs on a variety of datasets. L-LDA associates each label with one topic in direct correspondence, and is a natural extension of both LDA and multinomial Naïve Bayes. In our experiment, each document can have several labels, each corresponding to one of the 33 ACE event types. In this way, we can easily map the goal of predicting the possible events in a document into a multi-label classification problem.

## 5 Experiment

We set up two experiments to investigate the effect of topic information.

First, we did a 5-fold cross-validation on the whole ACE 2005 corpus. We report the overall Precision (P), Recall (R), and F-Measure (F).

Second, we did an experiment to address the crucial issue of mismatch in topic distribution between training and test corpora. In this experiment, the whole ACE 2005 corpus is used as the training data, and unfiltered New York Times newswire data (NYT) is used for testing. The NYT corpus comes from the same epoch (June 2003) as the ACE corpus, but there is no pre-selection. This test data contains 75 consecutive articles. We annotated the test data for the three most common event types in ACE –

*Attack*, *Die*, and *Meet* – and evaluated this balanced corpus on these three events.

### 5.1 Event Extraction Baseline System

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, because the definition of event mention argument constrains them to appear in the same sentence. The system combines pattern matching with statistical models. In the training process, for every event mention in the ACE training corpus, patterns are constructed based on the sequences of constituent heads separating the trigger and arguments. A set of Maximum Entropy based classifiers are also trained:

- **Argument Classifier:** to distinguish arguments of a potential trigger from non-arguments; uses local features like the event type of the potential trigger, path from the mention to the trigger, mention type, head word of the mention, etc.
- **Role Classifier:** to classify arguments by argument role; uses similar features as the argument classifier
- **Trigger Classifier:** Given local evidence, like the potential trigger word, the event type, and a set of arguments, to determine whether this 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 trigger classifier is applied to the potential event mention; if the result is successful, this event mention is reported<sup>3</sup>.

---

<sup>3</sup> Note that argument / role recall is rather low, because it is dependent on the correct recognition and classification of entity mentions, whose F measure (with our system) is about 81% for named mentions and lower for nominal and pronominal mentions.

## 5.2 Topic Features

Encoding topic features into the baseline system is straightforward: as the occurrence of an event is decided in the final classifier – the trigger classifier – we add topic features to this final classifier. Although the argument / role classifiers have already been applied, we can still improve the argument / role classification, because only when a word is tagged as a trigger will all the arguments/roles related to it be reported.

The unsupervised LDA was trained on the entire 2003 NYT newswire except for June to avoid overlap with the test data, a total of 27,827 articles; we choose  $K=30$ , which means we treat the whole corpus as a combination of 30 latent topics<sup>4</sup>.

The multi-label text classifier was trained on the same ACE training data as the event extraction, where each label corresponds to one event type, and there is an extra “none” tag when there are no events in the document. Thus, there are in total 34 labels.

For inference, we use the posterior Dirichlet parameters  $\gamma^*(w)$  associated with the document (Blei 2003) as our topic features, which is a fixed set of real-values. Thus, using the multi-label text classifier, there are 34 newly-added features; while using unsupervised LDA, there are 30 newly-added features. Stanford topic modeling software is used for both the multi-label text classifier and unsupervised LDA.

For preprocessing, we remove all words on a stop word list. Also, to reduce data sparseness, all inflected words are changed to their root form (e.g. “attackers”→“attacker”).

## 5.3 Evaluation on ACE data

We might expect supervised topic features to outperform unsupervised topic features, when the distribution of training and testing data are the same, because its correlation to event type is clearer and explicit. However, this turns out not to be true in our experiment (Table 2): the unsupervised features work better than the supervised features. This is understandable given that there are only hundreds of training documents for the supervised topic model, and the precision of the document classification is not very good, as we mentioned before in section 4.2. For unsupervised topics, we have a much larger corpus, and the topics extracted, although they

may not correspond directly to each event type, predicate a scenario where a specific event might occur.

## 5.4 Evaluation on NYT data

From the ACE evaluation, we can see that the unsupervised LDA works better than a supervised classifier, which indicates that even if the training and testing data are from the same distribution, the unsupervised topic features are more helpful. In our second evaluation, we evaluate on a more balanced newswire corpus, with no pre-selection.

First, we implement Grishman (2010)’s solution (Simple Combination) to combine the document event classifier (a bag-of-words maximum-entropy model) with local evidence used in the baseline system. The basic idea is that if a document is classified as not related to a specific event, it should not contain any such events; while if it is related, there should be such events. Thus, an event will be reported if

$$\sqrt{P(\text{reportable\_event}) \times P(\text{relevant\_document})} > \tau$$

where  $P(\text{reportable\_event})$  is the confidence score from the baseline system, while  $P(\text{relevant\_document})$  is computed from the document classifier.

Table 3 shows that the simple combination method (geometric mean of probabilities) performs a little better than baseline. However, we find that the gains are unevenly spread across different events. For *Attack* events, it provides some benefit (from 57.9% to 59.6% F score for trigger labeling), whereas for *Die* and *Meet* events it does not improve much. This might be because *Attack* events are closely tied to a document’s main topic, and using only the main topic can give a good prediction. But *Die* and *Meet* events are not closely tied to the document main topic, and so the simple combination does not help much.

Unsupervised LDA performs best of all, which indicates that the real distribution in the balanced corpus can provide useful guidance for event extraction, while supervised features might not provide enough information, especially when testing on a balanced corpus.

---

<sup>4</sup> We tested some other values of  $K$ , and found  $K=30$  works best, although we did not systematically explore alternative values.

Performance System	Trigger Classification			Argument Classification			Role Classification		
	P	R	F	P	R	F	P	R	F
Baseline system	64.3	51.1	56.9	69.4	21.8	33.2	62.8	19.7	30.0
Multi-label classifier	66.8	50.0	57.2	54.4	25.5	34.7	48.9	22.9	31.1
Unsupervised LDA	63.9	59.7	<b>61.7</b>	71.1	27.0	<b>39.1</b>	64.6	24.5	<b>35.5</b>

Table 2. Overall performance on ACE test data

Performance System	Trigger Classification			Argument Classification			Role Classification		
	P	R	F	P	R	F	P	R	F
Baseline system	53.8	51.1	52.4	41.4	19.7	26.7	39.4	18.8	25.4
Simple Combination	63.1	47.4	54.2	41.4	19.7	26.7	39.4	18.8	25.4
Multi-label classifier	60.8	65.7	63.2	35.6	27.9	31.3	31.9	25.0	28.0
Unsupervised LDA	60.3	81.0	<b>69.2</b>	45.3	34.6	<b>39.2</b>	44.0	33.7	<b>38.1</b>

Table 3. Performance on NYT collection

## 5.5 NYT Data Analysis

Here, we give some examples to show why topic information helps. First, we give an example where the supervised topics method does not work but unsupervised does. In our baseline system, many verbs in sports or other articles will be incorrectly tagged as *Attack* events. In such cases, as there are very few sports articles in ACE training data, and there is no event type related to sport, the supervised classifier might not capture this feature, and prefer to connect a sports article to an *Attack* event in the testing phase, because there are a lot of words like “shot”, “fight”. However, as there are a lot of sports articles in NYT data, the unsupervised LDA can capture this topic. Here is an example:

(2) *His only two **shots** of the game came in overtime and the goal was just his second of the playoffs, but it couldn't have been bigger.*

In the ACE training data, “shot” is tagged 67.5% of the time as an *Attack* event. We checked the data and found that there are very few sports articles in the ACE corpus, and the word “shot” never appears in these documents. Thus, a supervised classifier will prefer to tag a document containing the word “shot” as containing an *Attack* event. However, because a sports topic can be explicitly extracted from an

unannotated corpus that contains a reasonable portion of sports articles, the unsupervised model would be able to build a latent topic  $T$  which contains sports-related words like “racket”, “tennis”, “score” etc. Thus, most training documents which contain “shot” will have a low value of  $T$ ; while the sports documents (although very few), will have a high value of  $T$ . Thus, the system will see both a positive feature value (the word is “shot”), and a negative feature value ( $T$ 's value is high), and still has the chance to correctly tag this “shot” as *not-an-event*, while in the baseline system, the system will incorrectly tag it as an *Attack* event because there are only positive feature values.

The topic features can also help other event types. For *Die* events, consider:

(3) *A woman lay unconscious and **dying** at Suburban Hospital in Bethesda, Md.*

The word “dying” only appears 45.5% as a *Die* event in the training data, and is not tagged as a *Die* event by the baseline system. The reason is that there are a lot of metaphors that do not represent true *Die* events, like “dying nation”, “dying business”, “dying regime”. However, when connected to the latent topic features, we know that for some topics, we can confidently tag it as a *Die* event.

For *Meet* events, we also find cases where topic features help:

(4) *President Bush meets Tuesday with Arab leaders in Egypt and the next day with the Israeli and Palestinian prime ministers in Jordan,....*

The baseline system misses this *Meet* event. The word “meets” only appears 25% of the time as a *Meet* event in the training data, because there are phrases like “meets the requirement”, “meets the standard” which are not *Meet* events. However, adding topic features, we can correct this and similar event detection errors.

## 6 Conclusion

We proposed to use a topic model (LDA) to provide document level topic information for event trigger classification. The advantage of LDA for text classification or clustering is that it treats each document as a mixture of several topics instead of one, providing a more natural connection to the event extraction task. Both supervised and unsupervised LDA were applied. We evaluated the influence on two sets: one with the same distribution as the training data; the other a more balanced newswire collection without pre-selection.

Our experiments indicated that an unsupervised document-level topic model trained on a large corpus yields substantial improvements in extraction performance and is considerably more effective than a supervised topic model trained on a smaller annotated corpus.

## References

- David Ahn. 2006. *The stages of event extraction*. Proc. COLING/ACL 2006 Workshop on Annotating and Reasoning about Time and Events. Sydney, Australia.
- David Blei, Andrew Ng, and Michael Jordan. 2003. "Latent Dirichlet Allocation". *Journal of Machine Learning Research* 3: pp. 993–1022
- J. Finkel, T. Grenager, and C. Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 363–370, Ann Arbor, MI, June.
- Thomas Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences* 101 (Suppl. 1): 5228–5235. doi:10.1073/pnas.0307752101. PMID 14872004
- Ralph Grishman, David Westbrook and Adam Meyers. 2005. NYU’s English ACE 2005 System Description. In *Proc. ACE 2005 Evaluation Workshop, Gaithersburg, MD*.
- Ralph Grishman. 2010. The impact of task and corpus on Event Extraction Systems. In *Proceedings of LREC 2010*
- Heng Ji and Ralph Grishman. 2008. *Refining Event Extraction through Cross-Document Inference*. In *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, OH, June.
- Shasha Liao and Ralph Grishman. 2010b. Filtered Ranking for Bootstrapping in Event Extraction. In *Proceedings of COLING 2010*.
- Shasha Liao and Ralph Grishman. 2010a. Using Document Level Cross-Event Inference to Improve Event Extraction. In *Proceedings of ACL 2010*
- M. Maslennikov and T. Chua. 2007. *A Multi resolution Framework for Information Extraction from Free Text*. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 592–599, Prague, Czech Republic, June.
- S. Patwardhan and E. Riloff. 2007. *Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions*. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 717–727, Prague, Czech Republic, June.
- Ellen Riloff. 1996. Automatically Generating Extraction Patterns from Untagged Text. In *Proc. Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, 1996, pp. 1044-1049.
- Daniel Ramage, David Hall, Ramesh Nallapati, Christopher D. Manning 2009. Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*