In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)

Peeling Back the Layers: Detecting Event Role Fillers in Secondary Contexts

Ruihong Huang and Ellen Riloff
School of Computing
University of Utah
Salt Lake City, UT 84112
{huangrh,riloff}@cs.utah.edu

Abstract

The goal of our research is to improve event extraction by learning to identify secondary role filler contexts in the absence of event keywords. We propose a multi-layered event extraction architecture that progressively "zooms in" on relevant information. Our extraction model includes a document genre classifier to recognize event narratives, two types of sentence classifiers, and noun phrase classifiers to extract role fillers. These modules are organized as a pipeline to gradually zero in on event-related information. We present results on the MUC-4 event extraction data set and show that this model performs better than previous systems.

1 Introduction

Event extraction is an information extraction (IE) task that involves identifying the role fillers for events in a particular domain. For example, the Message Understanding Conferences (MUCs) challenged NLP researchers to create event extraction systems for domains such as terrorism (e.g., to identify the perpetrators, victims, and targets of terrorism events) and management succession (e.g., to identify the people and companies involved in corporate management changes).

Most event extraction systems use either a learning-based classifier to label words as role fillers, or lexico-syntactic patterns to extract role fillers from pattern contexts. Both approaches, however, generally tackle event recognition and role filler extraction at the same time. In other words, most event extraction systems primarily recognize contexts that explicitly refer to a relevant event. For example, a system that extracts information about murders will recognize expressions associated with murder (e.g., "killed", "assassinated", or "shot to death") and extract role fillers from the surrounding context. But many role fillers occur in contexts that do not explicitly mention the event, and those fillers are often overlooked. For example, the perpetrator of a murder may be mentioned in the context of an arrest, an eyewitness report, or speculation about possible suspects. Victims may be named in sentences that discuss the aftermath of the event, such as the identification of bodies, transportation of the injured to a hospital, or conclusions drawn from an investigation. We will refer to these types of sentences as "secondary contexts" because they are generally not part of the main event description. Discourse analysis is one option to explicitly link these secondary contexts to the event, but discourse modelling is itself a difficult problem.

The goal of our research is to improve event extraction by learning to identify secondary role filler contexts in the absence of event keywords. We create a set of classifiers to recognize *role-specific contexts* that suggest the presence of a likely role filler regardless of whether a relevant event is mentioned or not. For example, our model should recognize that a sentence describing an arrest probably includes a reference to a perpetrator, even though the crime itself is reported elsewhere.

Extracting information from these secondary contexts can be risky, however, unless we know that the larger context is discussing a relevant event. To

address this, we adopt a two-pronged strategy for event extraction that handles *event narrative* documents differently from other documents. We define an event narrative as an article whose main purpose is to report the details of an event. We apply the *role-specific sentence classifiers* only to event narratives to aggressively search for role fillers in these stories. However, other types of documents can mention relevant events too. The MUC-4 corpus, for example, includes interviews, speeches, and terrorist propaganda that contain information about terrorist events. We will refer to these documents as *fleeting reference* texts because they mention a relevant event somewhere in the document, albeit briefly. To ensure that relevant information is extracted from all documents, we also apply a conservative extraction process to every document to extract facts from explicit event sentences.

Our complete event extraction model, called TIER, incorporates both document genre and role-specific context recognition into 3 layers of analysis: document analysis, sentence analysis, and noun phrase (NP) analysis. At the top level, we train a text genre classifier to identify event narrative documents. At the middle level, we create two types of sentence classifiers. *Event sentence classifiers* identify sentences that are associated with relevant events, and *role-specific context classifiers* identify sentences that contain possible role fillers irrespective of whether an event is mentioned. At the lowest level, we use *role filler extractors* to label individual noun phrases as role fillers. As documents pass through the pipeline, they are analyzed at different levels of granularity. All documents pass through the event sentence classifier, and event sentences are given to the role filler extractors. Documents identified as event narratives additionally pass through role-specific sentence classifiers, and the role-specific sentences are also given to the role filler extractors. This multi-layered approach creates an event extraction system that can discover role fillers in a variety of different contexts, while maintaining good precision.

In the following sections, we position our research with respect to related work, present the details of our multi-layered event extraction model, and show experimental results for five event roles using the MUC-4 data set.

2 Related Work

Some event extraction data sets only include documents that describe relevant events (e.g., well-known data sets for the domains of corporate acquisitions (Freitag, 1998b; Freitag and McCallum, 2000; Finn and Kushmerick, 2004), job postings (Califf and Mooney, 2003; Freitag and McCallum, 2000), and seminar announcements (Freitag, 1998b; Ciravegna, 2001; Chieu and Ng, 2002; Finn and Kushmerick, 2004; Gu and Cercone, 2006). But many IE data sets present a more realistic task where the IE system must determine whether a relevant event is present in the document, and if so, extract its role fillers. Most of the Message Understanding Conference data sets represent this type of event extraction task, containing (roughly) a 50/50 mix of relevant and irrelevant documents (e.g., MUC-3, MUC-4, MUC-6, and MUC-7 (Hirschman, 1998)). Our research focuses on this setting where the event extraction system is not assured of getting only relevant documents to process.

Most event extraction models can be characterized as either pattern-based or classifier-based approaches. Early event extraction systems used handcrafted patterns (e.g., (Appelt et al., 1993; Lehnert et al., 1991)), but more recent systems generate patterns or rules automatically using supervised learning (e.g., (Kim and Moldovan, 1993; Riloff, 1993; Soderland et al., 1995; Huffman, 1996; Freitag, 1998b; Ciravegna, 2001; Califf and Mooney, 2003)), weakly supervised learning (e.g., (Riloff, 1996; Riloff and Jones, 1999; Yangarber et al., 2000; Sudo et al., 2003; Stevenson and Greenwood, 2005)), or unsupervised learning (e.g., (Shinyama and Sekine, 2006; Sekine, 2006)). In addition, many classifiers have been created to sequentially label event role fillers in a sentence (e.g., (Freitag, 1998a; Chieu and Ng, 2002; Finn and Kushmerick, 2004; Li et al., 2005; Yu et al., 2005)). Research has also been done on relation extraction (e.g., (Roth and Yih, 2001; Zelenko et al., 2003; Bunescu and Mooney, 2007)), but that task is different from event extraction because it focuses on isolated relations rather than template-based event analysis.

Most event extraction systems scan a text and search small context windows using patterns or a classifier. However, recent work has begun to ex-

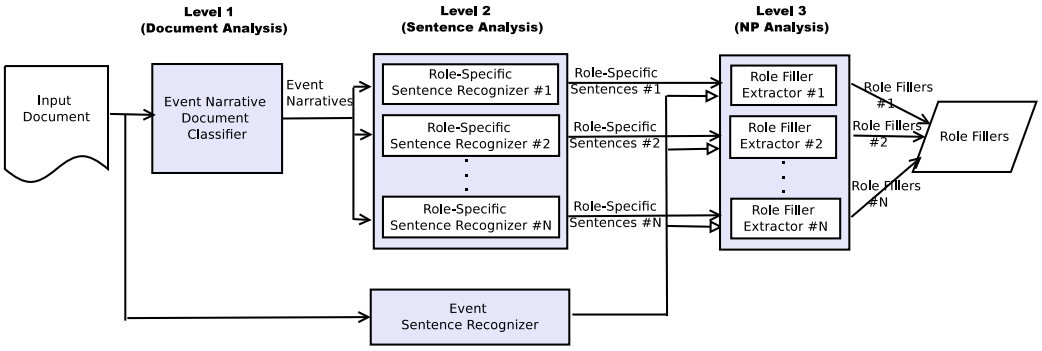


Figure 1: TIER: A Multi-Layered Architecture for Event Extraction

plore more global approaches. (Maslennikov and Chua, 2007) use discourse trees and local syntactic dependencies in a pattern-based framework to incorporate wider context. Ji and Grishman (2008) enforce event role consistency across different documents. (Liao and Grishman, 2010) use cross-event inference to help with the extraction of role fillers shared across events. And there have been several recent IE models that explore the idea of identifying relevant sentences to gain a wider contextual view and then extracting role fillers. (Gu and Cercone, 2006) created HMMs to first identify relevant sentences, but their research focused on eliminating redundant extractions and worked with seminar announcements, where the system was only given relevant documents. (Patwardhan and Riloff, 2007) developed a system that learns to recognize event sentences and uses patterns that have a *semantic affinity* for an event role to extract role fillers. GLACIER (Patwardhan and Riloff, 2009) jointly considers sentential evidence and phrasal evidence in a unified probabilistic framework. Our research follows in the same spirit as these approaches by performing multiple levels of text analysis. But our event extraction model includes two novel contributions: (1) we develop a set of *role-specific* sentence classifiers to learn to recognize *secondary contexts* associated with each type of event role , and (2) we exploit text genre to incorporate a third level of analysis that enables the system to aggressively hunt for role fillers in documents that are event narratives. In Section 5, we compare the performance of our model with both the GLACIER system and Patwardhan & Riloff's semantic affinity model.

3 A Multi-Layered Approach to Event Extraction

The main idea behind our approach is to analyze documents at multiple levels of granularity in order to identify role fillers that occur in different types of contexts. Our event extraction model progressively "zooms in" on relevant information by first identifying the document type, then identifying sentences that are likely to contain relevant information, and finally analyzing individual noun phrases to identify role fillers. The key advantage of this architecture is that it allows us to search for information using two different principles: (1) we look for contexts that directly refer to the event, as per most traditional event extraction systems, and (2) we look for secondary contexts that are often associated with a specific type of role filler. Identifying these *role-specific contexts* can root out important facts would have been otherwise missed. Figure 1 shows the multi-layered pipeline of our event extraction system.

An important aspect of our model is that two different strategies are employed to handle documents of different types. The event extraction task is to find any description of a relevant event, even if the event is not the topic of the article.[1] Consequently, all documents are given to the event sentence recognizers and their mission is to identify any sentence that mentions a relevant event. This path through the pipeline is conservative because information is extracted only from event sentences, but all documents are processed, including stories that contain only a fleeting reference to a relevant event.

[1]Per the MUC-4 task definition (MUC-4 Proceedings, 1992).

The second path through the pipeline performs additional processing for documents that belong to the event narrative text genre. For event narratives, we assume that most of the document discusses a relevant event so we can more aggressively hunt for event-related information in secondary contexts.

In this section, we explain how we create the two types of sentence classifiers and the role filler extractors. We will return to the issue of document genre and the event narrative classifier in Section 4.

3.1 Sentence Classification

We have argued that event role fillers commonly occur in two types of contexts: event contexts and role-specific secondary contexts. For the purposes of this research, we use sentences as our definition of a "context", although there are obviously many other possible definitions. An *event context* is a sentence that describes the actual event. A *secondary context* is a sentence that provides information related to an event but in the context of other activities that precede or follow the event.

For both types of classifiers, we use exactly the same feature set, but we train them in different ways. The MUC-4 corpus used in our experiments includes a training set consisting of documents and answer keys. Each document that describes a relevant event has answer key templates with the role fillers (*answer key strings*) for each event. To train the event sentence recognizer, we consider a sentence to be a positive training instance if it contains one or more answer key strings from any of the event roles. This produced 3,092 positive training sentences. All remaining sentences that do not contain any answer key strings are used as negative instances. This produced 19,313 negative training sentences, yielding a roughly 6:1 ratio of negative to positive instances.

There is no guarantee that a classifier trained in this way will identify event sentences, but our hypothesis was that training across all of the event roles together would produce a classifier that learns to recognize general event contexts. This approach was also used to train GLACIER's sentential event recognizer (Patwardhan and Riloff, 2009), and they demonstrated that this approach worked reasonably well when compared to training with event sentences labelled by human judges.

The main contribution of our work is introducing additional *role-specific sentence classifiers* to seek out role fillers that appear in less obvious secondary contexts. We train a set of role-specific sentence classifiers, one for each type of event role. Every sentence that contains a role filler of the appropriate type is used as a positive training instance. Sentences that do not contain any answer key strings are negative instances.[2] In this way, we force each classifier to focus on the contexts specific to its particular event role. We expect the role-specific sentence classifiers to find some secondary contexts that the event sentence classifier will miss, although some sentences may be classified as both.

Using all possible negative instances would produce an extremely skewed ratio of negative to positive instances. To control the skew and keep the training set-up consistent with the event sentence classifier, we randomly choose from the negative instances to produce a 6:1 ratio of negative to positive instances.

Both types of classifiers use an SVM model created with SVMlin (Keerthi and DeCoste, 2005), and exactly the same features. The feature set consists of the unigrams and bigrams that appear in the training texts, the semantic class of each noun phrase[3], plus a few additional features to represent the tense of the main verb phrase in the sentence and whether the document is long (> 35 words) or short (< 5 words). All of the feature values are binary.

3.2 Role Filler Extractors

Our extraction model also includes a set of role filler extractors, one per event role. Each extractor receives a sentence as input and determines which noun phrases (NPs) in the sentence are fillers for the event role. To train an SVM classifier, noun phrases corresponding to answer key strings for the event role are positive instances. We randomly choose among all noun phrases that are not in the answer keys to create a 10:1 ratio of negative to positive instances.

[2] We intentionally do not use sentences that contain fillers for competing event roles as negative instances because sentences often contain multiple role fillers of different types (e.g., a weapon may be found near a body). Sentences without any role fillers are certain to be irrelevant contexts.

[3] We used the Sundance parser (Riloff and Phillips, 2004) to identify noun phrases and assign semantic class labels.

The feature set for the role filler extractors is much richer than that of the sentence classifiers because they must carefully consider the local context surrounding a noun phrase. We will refer to the noun phrase being labelled as the *targeted NP*. The role filler extractors use three types of features:

Lexical features: we represent four words to the left and four words to the right of the targeted NP, as well as the head noun and modifiers (adjectives and noun modifiers) of the targeted NP itself.

Lexico-syntactic patterns: we use the AutoSlog pattern generator (Riloff, 1993) to automatically create lexico-syntactic patterns around each noun phrase in the sentence. These patterns are similar to dependency relations in that they typically represent the syntactic role of the NP with respect to other constituents (e.g., subject-of, object-of, and noun arguments).

Semantic features: we use the Stanford NER tagger (Finkel et al., 2005) to determine if the targeted NP is a named entity, and we use the Sundance parser (Riloff and Phillips, 2004) to assign semantic class labels to each NP's head noun.

4 Event Narrative Document Classification

One of our goals was to explore the use of *document genre* to permit more aggressive strategies for extracting role fillers. In this section, we first present an analysis of the MUC-4 data set which reveals the distribution of event narratives in the corpus, and then explain how we train a classifier to automatically identify event narrative stories.

4.1 Manual Analysis

We define an *event narrative* as an article whose main focus is on reporting the details of an event. For the purposes of this research, we are only concerned with events that are relevant to the event extraction task (i.e., terrorism). An *irrelevant document* is an article that does not mention any relevant events. In between these extremes is another category of documents that briefly mention a relevant event, but the event is not the focus of the article. We will refer to these documents as *fleeting reference* documents. Many of the fleeting reference documents in the MUC-4 corpus are transcripts of interviews, speeches, or terrorist propaganda communiques that refer to a terrorist event and mention at least one role filler, but within a discussion about a different topic (e.g., the political ramifications of a terrorist incident).

To gain a better understanding of how we might create a system to automatically distinguish event narrative documents from fleeting reference documents, we manually labelled the 116 relevant documents in our tuning set. This was an informal study solely to help us understand the nature of these texts.

	# of Event Narratives	# of Fleeting Ref. Docs	Acc
Gold Standard	54	62	
Heuristics	40	55	.82

Table 1: Manual Analysis of Document Types

The first row of Table 1 shows the distribution of event narratives and fleeting references based on our "gold standard" manual annotations. We see that more than half of the relevant documents (62/116) are *not* focused on reporting a terrorist event, even though they contain information about a terrorist event somewhere in the document.

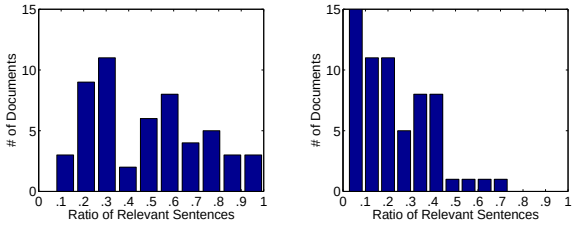
4.2 Heuristics for Event Narrative Identification

Our goal is to train a document classifier to automatically identify event narratives. The MUC-4 answer keys reveal which documents are relevant and irrelevant with respect to the terrorism domain, but they do not tell us which relevant documents are event narratives and which are fleeting reference stories. Based on our manual analysis of the tuning set, we developed several heuristics to help separate them.

We observed two types of clues: the location of the relevant information, and the density of relevant information. First, we noticed that event narratives tend to mention relevant information within the first several sentences, whereas fleeting reference texts usually mention relevant information only in the middle or end of the document. Therefore our first heuristic requires that an event narrative mention a role filler within the first 7 sentences.

Second, event narratives generally have a higher density of relevant information. We use several criteria to estimate information density because a single criterion was inadequate to cover different sce-

narios. For example, some documents mention role fillers throughout the document. Other documents contain a high concentration of role fillers in some parts of the document but no role fillers in other parts. We developed three density heuristics to account for different situations. All of these heuristics count distinct role fillers. The first density heuristic requires that more than 50% of the sentences contain at least one role filler ($\frac{|RelSents|}{|AllSents|} > 0.5$) . Figure 2 shows histograms for different values of this ratio in the event narrative (a) vs. the fleeting reference documents (b). The histograms clearly show that documents with a high ($> 50\%$) ratio are almost always event narratives.



(a) (b)

Figure 2: Histograms of Density Heuristic #1 in Event Narratives (a) vs. Fleeting References (b).

A second density heuristic requires that the ratio of different *types* of roles filled to sentences be $>$ 50% ($\frac{|Roles|}{|AllSents|} > 0.5$). A third density heuristic requires that the ratio of distinct role *fillers* to sentences be $> 70\%$ ($\frac{|RoleFillers|}{|AllSents|} > 0.7$). If any of these three criteria are satisfied, then the document is considered to have a high density of relevant information.[4]

We use these heuristics to label a document as an event narrative if: (1) it has a high density of relevant information, and (2) it mentions a role filler within the first 7 sentences.

The second row of Table 1 shows the performance of these heuristics on the tuning set. The heuristics correctly identify $\frac{40}{54}$ event narratives and $\frac{55}{62}$ fleeting reference stories, to achieve an overall accuracy of 82%. These results are undoubtedly optimistic because the heuristics were derived from analysis of the tuning set. But we felt confident enough to move forward with using these heuristics to generate training data for an event narrative classifier.

[4]Heuristic #1 covers most of the event narratives.

4.3 Event Narrative Classifier

The heuristics above use the answer keys to help determine whether a story belongs to the event narrative genre, but our goal is to create a classifier that can identify event narrative documents without the benefit of answer keys. So we used the heuristics to automatically create training data for a classifier by labelling each relevant document in the training set as an event narrative or a fleeting reference document. Of the 700 relevant documents, 292 were labeled as event narratives. We then trained a document classifier using the 292 event narrative documents as positive instances and all irrelevent training documents as negative instances. The 308 relevant documents that were not identified as event narratives were discarded to minimize noise (i.e., we estimate that our heuristics fail to identify 25% of the event narratives). We then trained an SVM classifier using bag-of-words (unigram) features.

Table 2 shows the performance of the event narrative classifier on the manually labeled tuning set. The classifier identified 69% of the event narratives with 63% precision. Overall accuracy was 81%.

Recall	Precision	Accuracy
.69	.63	.81

Table 2: Event Narrative Classifier Results

At first glance, the performance of this classifier is mediocre. However, these results should be interpreted loosely because there is not always a clear dividing line between event narratives and other documents. For example, some documents begin with a specific event description in the first few paragraphs but then digress to discuss other topics. Fortunately, it is not essential for TIER to have a perfect event narrative classifier since all documents will be processed by the event sentence recognizer anyway. The recall of the event narrative classifier means that nearly 70% of the event narratives will get additional scrutiny, which should help to find additional role fillers. Its precision of 63% means that some documents that are not event narratives will also get additional scrutiny, but information will be extracted only if both the role-specific sentence recognizer and NP extractors believe they have found

Method	PerpInd	PerpOrg	Target	Victim	Weapon	Average
Baselines						
AutoSlog-TS	33/49/40	52/33/41	54/59/56	49/54/51	38/44/41	45/48/46
Semantic Affinity	48/39/43	36/58/45	56/46/50	46/44/45	53/46/50	48/47/47
GLACIER	51/58/**54**	34/45/38	43/72/53	55/58/56	57/53/55	48/57/52
New Results without document classification						
AllSent	25/67/36	26/78/39	34/83/49	32/72/45	30/75/43	30/75/42
EventSent	52/54/53	50/44/47	52/67/59	55/51/53	56/57/56	53/54/54
RoleSent	37/54/44	37/58/45	49/75/59	52/60/55	38/66/48	43/63/51
EventSent+RoleSent	38/60/46	36/63/46	47/78/59	52/64/57	36/66/47	42/66/51
New Results with document classification						
DomDoc/EventSent+DomDoc/RoleSent	45/54/49	42/51/46	51/68/58	54/56/55	46/63/53	48/58/52
EventSent+DomDoc/RoleSent	43/59/50	45/61/**52**	51/77/**61**	52/61/56	44/66/53	47/65/54
EventSent+ENarrDoc/RoleSent	48/57/52	46/53/50	51/73/60	56/60/**58**	53/64/**58**	51/62/**56**

Table 3: Experimental results, reported as Precision/Recall/F-score

something relevant.

4.4 Domain-relevant Document Classifier

For comparison's sake, we also created a document classifier to identify *domain-relevant* documents. That is, we trained a classifier to determine whether a document is relevant to the domain of terrorism, irrespective of the style of the document. We trained an SVM classifier with the same bag-of-words feature set, using all relevant documents in the training set as positive instances and all irrelevant documents as negative instances. We use this classifier for several experiments described in the next section.

5 Evaluation

5.1 Data Set and Metrics

We evaluated our approach on a standard benchmark collection for event extraction systems, the MUC-4 data set (MUC-4 Proceedings, 1992). The MUC-4 corpus consists of 1700 documents with associated answer key templates. To be consistent with previously reported results on this data set, we use the 1300 DEV documents for training, 200 documents (TST1+TST2) as a tuning set and 200 documents (TST3+TST4) as the test set. Roughly half of the documents are relevant (i.e., they mention at least 1 terrorist event) and the rest are irrelevant.

We evaluate our system on the five MUC-4 "string-fill" event roles: *perpetrator individuals*, *perpetrator organizations*, *physical targets*, *victims* and *weapons*. The complete IE task involves template generation, which is complex because many documents have multiple templates (i.e., they discuss multiple events). Our work focuses on extracting individual facts and not on template generation per se (e.g., we do not perform coreference resolution or event tracking). Consequently, our evaluation follows that of other recent work and evaluates the accuracy of the extractions themselves by matching the head nouns of extracted NPs with the head nouns of answer key strings (e.g., "armed guerrillas" is considered to match "guerrillas")[5]. Our results are reported as Precision/Recall/F(1)-score for each event role separately. We also show an overall average for all event roles combined.[6]

5.2 Baselines

As baselines, we compare the performance of our IE system with three other event extraction systems. The first baseline is AutoSlog-TS (Riloff, 1996), which uses domain-specific extraction patterns. AutoSlog-TS applies its patterns to every sentence in every document, so does not attempt to explicitly identify relevant sentences or documents. The next two baselines are more recent systems: the (Patwardhan and Riloff, 2007) *semantic affinity* model and the (Patwardhan and Riloff, 2009) GLACIER system. The *semantic affinity* approach

[5]Pronouns were discarded since we do not perform coreference resolution. Duplicate extractions with the same head noun were counted as one hit or one miss.

[6]We generated the Average scores ourselves by macro-averaging over the scores reported for the individual event roles.

explicitly identifies event sentences and uses patterns that have a semantic affinity for an event role to extract role fillers. GLACIER is a probabilistic model that incorporates both phrasal and sentential evidence jointly to label role fillers.

The first 3 rows in Table 3 show the results for each of these systems on the MUC-4 data set. They all used the same evaluation criteria as our results.

5.3 Experimental Results

The lower portion of Table 3 shows the results of a variety of event extraction models that we created using different components of our system. The **AllSent** row shows the performance of our Role Filler Extractors when applied to every sentence in every document. This system produced high recall, but precision was consistently low.

The **EventSent** row shows the performance of our Role Filler Extractors applied only to the *event sentences* identified by our event sentence classifier. This boosts precision across all event roles, but with a sharp reduction in recall. We see a roughly 20 point swing from recall to precision. These results are similar to GLACIER's results on most event roles, which isn't surprising because GLACIER also incorporates event sentence identification.

The **RoleSent** row shows the results of our Role Filler Extractors applied only to the *role-specific sentences* identified by our classifiers. We see a 12-13 point swing from recall to precision compared to the **AllSent** row. This result is consistent with our hypothesis that many role fillers exist in role-specific contexts that are not event sentences. As expected, extracting facts from role-specific contexts that do not necessarily refer to an event is less reliable. The **EventSent+RoleSent** row shows the results when information is extracted from both types of sentences. We see slightly higher recall, which confirms that one set of extractions is not a strict subset of the other, but precision is still relatively low.

The next set of experiments incorporates document classification as the third layer of text analysis. The **DomDoc/EventSent+DomDoc/RoleSent** row shows the results of applying both types of sentence classifiers only to documents identified as domain-relevant by the Domain-relevant Document (**DomDoc**) Classifier described in Section 4.4. Extracting information only from domain-relevant documents improves precision by $+6$, but also sacrifices 8 points of recall.

The **EventSent** row reveals that information found in event sentences has the highest precision, even without relying on document classification. We concluded that evidence of an event sentence is probably sufficient to warrant role filler extraction irrespective of the style of the document. As we discussed in Section 4, many documents contain only a fleeting reference to an event, so it is important to be able to extract information from those isolated event descriptions as well. Consequently, we created a system, **EventSent+DomDoc/RoleSent**, that extracts information from event sentences in *all* documents, but extracts information from role-specific sentences only if they appear in a domain-relevant document. This architecture captured the best of both worlds: recall improved from 58% to 65% with only a one point drop in precision.

Finally, we evaluated the idea of using document *genre* as a filter instead of domain relevance. The last row, **EventSent+ENarrDoc/RoleSent**, shows the results of our final architecture which extracts information from event sentences in all documents, but extracts information from role-specific sentences only in Event Narrative documents. This architecture produced the best F1 score of 56. This model increases precision by an additional 4 points and produces the best balance of recall and precision.

Overall, TIER's multi-layered extraction architecture produced higher F1 scores than previous systems on four of the five event roles. The improved recall is due to the additional extractions from secondary contexts. The improved precision comes from our two-pronged strategy of treating event narratives differently from other documents. TIER aggressively searches for extractions in event narrative stories but is conservative and extracts information only from event sentences in all other documents.

5.4 Analysis

We looked through some examples of TIER's output to try to gain insight about its strengths and limitations. TIER's role-specific sentence classifiers did correctly identify some sentences containing role fillers that were not classified as event sentences. Several examples are shown below, with the role

fillers in italics:

(1) "The victims were identified as *David Lecky*, director of the Columbus school, and *James Arthur Donnelly*."

(2) "There were *seven children*, including *four of the Vice President's children*, in the home at the time."

(3) "*The woman* fled and sought refuge inside the facilities of the Salvadoran Alberto Masferrer University, where she took a group of *students* as hostages, threatening them with *hand grenades*."

(4) "The FMLN stated that *several homes* were damaged and that animals were killed in the surrounding hamlets and villages."

The first two sentences identify victims, but the terrorist event itself was mentioned earlier in the document. The third sentence contains a perpetrator (*the woman*), victims (*students*), and weapons (*hand grenades*) in the context of a hostage situation after the main event (a bus attack), when the perpetrator escaped. The fourth sentence describes incidental damage to civilian homes following clashes between government forces and guerrillas.

However there is substantial room for improvement in each of TIER's subcomponents, and many role fillers are still overlooked. One reason is that it can be difficult to recognize acts of terrorism. Many sentences refer to a potentially relevant subevent (e.g., injury or physical damage) but recognizing that the event is part of a terrorist incident depends on the larger discourse. For example, consider the examples below that TIER did not recognize as relevant sentences:

(5) "Later, *two individuals* in a Chevrolet Opala automobile pointed AK rifles at the students, fired some shots, and quickly drove away."

(6) "Meanwhile, national police members who were dressed in civilian clothes seized university students *Hugo Martinez* and *Raul Ramirez*, who are still missing."

(7) "*All labor union offices* in San Salvador were looted."

In the first sentence, the event is described as someone pointing rifles at people and the perpetrators are referred to simply as individuals. There are no strong keywords in this sentence that reveal this is a terrorist attack. In the second sentence, police are being accused of state-sponsored terrorism when they seize civilians. The verb "seize" is common in this corpus, but usually refers to the seizing of weapons or drug stashes, not people. The third sentence describes a looting subevent. Acts of looting and vandalism are not usually considered to be terrorism, but in this article it is in the context of accusations of terrorist acts by government officials.

6 Conclusions

We have presented a new approach to event extraction that uses three levels of analysis: document genre classification to identify event narrative stories, two types of sentence classifiers, and noun phrase classifiers. A key contribution of our work is the creation of role-specific sentence classifiers that can detect role fillers in secondary contexts that do not directly refer to the event. Another important aspect of our approach is a two-pronged strategy that handles event narratives differently from other documents. TIER aggressively hunts for role fillers in event narratives, but is conservative about extracting information from other documents. This strategy produced improvements in both recall and precision over previous state-of-the-art systems.

This work just scratches the surface of using document genre identification to improve information extraction accuracy. In future work, we hope to identify additional types of document genre styles and incorporate genre directly into the extraction model. Coreference resolution and discourse analysis will also be important to further improve event extraction performance.

7 Acknowledgments

We gratefully acknowledge the support of the National Science Foundation under grant IIS-1018314 and the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0172. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, AFRL, or the U.S. government.

References

D. Appelt, J. Hobbs, J. Bear, D. Israel, and M. Tyson. 1993. FASTUS: a finite-state processor for information extraction from real-world text. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*.

R. Bunescu and R. Mooney. 2007. Learning to Extract Relations from the Web using Minimal Supervision. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

M.E. Califf and R. Mooney. 2003. Bottom-up Relational Learning of Pattern Matching rules for Information Extraction. *Journal of Machine Learning Research*, 4:177–210.

H.L. Chieu and H.T. Ng. 2002. A Maximum Entropy Approach to Information Extraction from Semi-Structured and Free Text. In *Proceedings of the 18th National Conference on Artificial Intelligence*.

F. Ciravegna. 2001. Adaptive Information Extraction from Text by Rule Induction and Generalisation. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*.

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.

A. Finn and N. Kushmerick. 2004. Multi-level Boundary Classification for Information Extraction. In *In Proceedings of the 15th European Conference on Machine Learning*, pages 111–122, Pisa, Italy, September.

D. Freitag and A. McCallum. 2000. Information Extraction with HMM Structures Learned by Stochastic Optimization. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 584–589, Austin, TX, August.

Dayne Freitag. 1998a. Multistrategy Learning for Information Extraction. In *Proceedings of the Fifteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers.

Dayne Freitag. 1998b. Toward General-Purpose Learning for Information Extraction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*.

Z. Gu and N. Cercone. 2006. Segment-Based Hidden Markov Models for Information Extraction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 481–488, Sydney, Australia, July.

L. Hirschman. 1998. "The Evolution of Evaluation: Lessons from the Message Understanding Conferences. *Computer Speech and Language*, 12.

S. Huffman. 1996. Learning Information Extraction Patterns from Examples. In Stefan Wermter, Ellen Riloff, and Gabriele Scheler, editors, *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, pages 246–260. Springer-Verlag, Berlin.

H. Ji and R. Grishman. 2008. Refining Event Extraction through Cross-Document Inference. In *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, OH, June.

S. Keerthi and D. DeCoste. 2005. A Modified Finite Newton Method for Fast Solution of Large Scale Linear SVMs. *Journal of Machine Learning Research*.

J. Kim and D. Moldovan. 1993. Acquisition of Semantic Patterns for Information Extraction from Corpora. In *Proceedings of the Ninth IEEE Conference on Artificial Intelligence for Applications*, pages 171–176, Los Alamitos, CA. IEEE Computer Society Press.

W. Lehnert, C. Cardie, D. Fisher, E. Riloff, and R. Williams. 1991. University of Massachusetts: Description of the CIRCUS System as Used for MUC-3. In *Proceedings of the Third Message Understanding Conference (MUC-3)*, pages 223–233, San Mateo, CA. Morgan Kaufmann.

Y. Li, K. Bontcheva, and H. Cunningham. 2005. Using Uneven Margins SVM and Perceptron for Information Extraction. In *Proceedings of Ninth Conference on Computational Natural Language Learning*, pages 72–79, Ann Arbor, MI, June.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48st Annual Meeting on Association for Computational Linguistics (ACL-10)*.

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 for Computational Linguistics*.

MUC-4 Proceedings. 1992. *Proceedings of the Fourth Message Understanding Conference (MUC-4)*. Morgan Kaufmann.

S. Patwardhan and E. Riloff. 2007. Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions. In *Proceedings of 2007 the Conference on Empirical Methods in Natural Language Processing (EMNLP-2007)*.

S. Patwardhan and E. Riloff. 2009. A Unified Model of Phrasal and Sentential Evidence for Information Extraction. In *Proceedings of 2009 the Conference on Empirical Methods in Natural Language Processing (EMNLP-2009)*.

E. Riloff and R. Jones. 1999. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*.

E. Riloff and W. Phillips. 2004. An Introduction to the Sundance and AutoSlog Systems. Technical Report UUCS-04-015, School of Computing, University of Utah.

E. Riloff. 1993. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of the 11th National Conference on Artificial Intelligence*.

E. Riloff. 1996. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1044–1049. The AAAI Press/MIT Press.

D. Roth and W. Yih. 2001. Relational Learning via Propositional Algorithms: An Information Extraction Case Study. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 1257–1263, Seattle, WA, August.

Satoshi Sekine. 2006. On-demand information extraction. In *Proceedings of Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING/ACL-06*.

Y. Shinyama and S. Sekine. 2006. Preemptive Information Extraction using Unrestricted Relation Discovery. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 304–311, New York City, NY, June.

S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. 1995. CRYSTAL: Inducing a conceptual dictionary. In *Proc. of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1314–1319.

M. Stevenson and M. Greenwood. 2005. A Semantic Approach to IE Pattern Induction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 379–386, Ann Arbor, MI, June.

K. Sudo, S. Sekine, and R. Grishman. 2003. An Improved Extraction Pattern Representation Model for Automatic IE Pattern Acquisition. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*.

R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. 2000. Automatic Acquisition of Domain Knowledge for Information Extraction. In *Proceedings of the Eighteenth International Conference on Computational Linguistics (COLING 2000)*.

K. Yu, G. Guan, and M. Zhou. 2005. Resumé Information Extraction with Cascaded Hybrid Model. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 499–506, Ann Arbor, MI, June.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel Methods for Relation Extraction. *Journal of Machine Learning Research*, 3.