# IMPROVING INFORMATION EXTRACTION BY DISCOURSE-GUIDED AND MULTIFACETED EVENT RECOGNITION

by

Ruihong Huang

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

School of Computing

The University of Utah

December 2014

# The University of Utah Graduate School

## STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Ruihong Huang**

has been approved by the following supervisory committee members:

| | | |
|---|---|---|
| **Ellen Riloff** | , Chair | **08/26/2014** Date Approved |
| **Jur van den Berg** | , Member | **08/25/2014** Date Approved |
| **Raymond Mooney** | , Member | **06/19/2014** Date Approved |
| **William Thompson** | , Member | **06/19/2014** Date Approved |
| **Suresh Venkatasubramanian** | , Member | **06/19/2014** Date Approved |

and by **Ross Whitaker**, Chair/Dean of

the Department/College/School of **School of Computing**

and by David B. Kieda, Dean of The Graduate School.

# ABSTRACT

Events are one important type of information throughout text. Event extraction is an information extraction (IE) task that involves identifying entities and objects (mainly noun phrases) that represent important roles in events of a particular type. However, the extraction performance of current event extraction systems is limited because they mainly consider local context (mostly isolated sentences) when making each extraction decision. My research aims to improve both coverage and accuracy of event extraction performance by explicitly identifying event contexts before extracting individual facts.

First, I introduce new event extraction architectures that incorporate discourse information across a document to seek out and validate pieces of event descriptions within the document. *TIER* is a multilayered event extraction architecture that performs text analysis at multiple granularities to progressively "zoom in" on relevant event information. *LINKER* is a unified discourse-guided approach that includes a structured sentence classifier to sequentially read a story and determine which sentences contain event information based on both the local and preceding contexts. Experimental results on two distinct event domains show that compared to previous event extraction systems, *TIER* can find more event information while maintaining a good extraction accuracy, and *LINKER* can further improve extraction accuracy.

Finding documents that describe a specific type of event is also highly challenging because of the wide variety and ambiguity of event expressions. In this dissertation, I present the multifaceted event recognition approach that uses event defining characteristics (facets), in addition to event expressions, to effectively resolve the complexity of event descriptions. I also present a novel bootstrapping algorithm to automatically learn event expressions as well as facets of events, which requires minimal human supervision. Experimental results show that the multifaceted event recognition approach can effectively identify documents that describe a particular type of event and make event extraction systems more precise.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ACKNOWLEDGEMENTS

I could never have enjoyed my last six years so much without the guidance, support, and efforts of so many people with whom I have interacted.

First, I am extremely grateful to have been under the guidance of my great advisor, Ellen Riloff, who has demonstrated to me how to be an excellent researcher, advisor, teacher, and friend. I have learned from her not just research and thinking skills, but also principles that I want to follow in my future life.

I would like to thank Hal Daumé for his guidance on my committee during my early stages of research. Thanks for showing me the rich world of statistical NLP. I would also like to thank all of my committee members, Jur van den Berg, Raymond Mooney, William Thompson, and Suresh Venkatasubramanian, for supervising my dissertation research and providing valuable input. I will always be grateful to my MS thesis advisor, Le Sun, for introducing me to the field. I am also grateful to Ryan Gabbard and Marjorie Freedman for mentoring me during my summer internship at Raytheon BBN Technologies.

I have been so lucky to work in the wonderful NLP Research Group at the University of Utah, surrounded by many great colleagues. With a ton of fresh and encouraging discussions on research and other fun, I have enjoyed working here every single day. Special thanks go to Siddharth Patwardhan, Nathan Gilbert, Sean Igo, Piyush Rai, Amit Goyal, jagadeesh jagarlamudi, Arvind Agarwal, Jiarong Jiang, Adam Teichert, Youngjun Kim, Ashequl Qadir, Lalindra de Silva, and Haibo Ding. Even more thanks to Nathan Gilbert, Sean Igo, Lalindra de Silva, and Ashequl Qadir, for your help with the annotations that are necessary in this dissertation. Without your time investment, patience, and hard work, this dissertation would have been incomplete.

I would also like to mention my dear friends, Qian, Shan, Yu, Yan, Shuying, and my wonderful roommate, Kali, for frequently teaching me the real wisdom of life, and with whom I shared so many vivid memories in Utah.

Many thanks to the staff at the School of Computing in the University of Utah. Special thanks go to Karen Feinauer and Ann Carlstrom for your immense support, so that I could entirely concentrate on my research.

My husband, Qin, my parents, Shaoguang and Lanfang, and my brother, Yanjun, have always supported me in everything I have chosen to do. This dissertation would have been impossible to complete in the absence of your much needed support.

Last, but not the least, I gratefully acknowledge the funding sources that supported this research. This research was supported in part by the National Science Foundation under grant IIS-1018314, the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0172, and the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI / NBC) contract number D12PC00285. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views, findings, and conclusions or recommendations contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBE, NSF, DARPA, AFRL, or the U.S. Government.

# CHAPTER 1

# INTRODUCTION

Events are one important type of information throughout text. Accurately extracting significant events from large volumes of text informs the government, companies, and the public regarding possible changing circumstances caused or implied by events.

Event extraction is an information extraction (IE) task that involves identifying entities and objects (mainly noun phrases) that represent important roles in events of a particular type. The extracted noun phrases are called role fillers of events and they are the participants of events, objects that are involved in events, or properties associated with an event. For example, event extraction systems for the terrorism domain identify the perpetrators, victims, and targets of terrorism events while systems for the management succession domain identify the people and companies involved in corporate management changes.

However, extracting event information completely and accurately is challenging mainly due to the high complexity of discourse phenomena. While this task has been studied over the last decades, the performance of current event extraction systems is limited because they mainly consider local context (mostly isolated sentences) and ignore the influences of wider contexts from the discourse. My research aims to improve both coverage and accuracy of event extraction performance by exploring discourse-guided models. By incorporating discourse information beyond an individual sentence, the discourse-guided models will seek out event information that tends to be overlooked by current event extraction systems and filter out extractions that seem to be valid when viewed locally.

Finding documents that describe a specific type of event is also challenging because of the wide variety and ambiguity of event expressions. My research also aims to accurately identify event relevant documents by proposing multifaceted event recognition. Event facets represent event defining characteristics. Multifaceted event recognition uses event facets, in addition to event expressions, to effectively resolve the complexity of event descriptions.

## 1.1 Discourse-Guided Event Extraction

Most current event extraction systems heavily rely on local contexts and individual event expressions when making extraction decisions. They primarily recognize contexts that explicitly refer to a relevant event and extract the noun phrases in those contexts as role fillers. 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. However, lacking the view of wider context limits the performance of traditional event extraction systems in two aspects.

First, the coverage of event extraction systems is limited because many role fillers occur in contexts that do not explicitly mention the event, and those fillers are often overlooked by current event extraction systems. 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. I will refer to these types of sentences as "secondary contexts" because they are generally not part of the main event description ("primary contexts"). Role fillers in secondary contexts are generally overlooked by current event extraction systems.

However, extracting information from these secondary contexts indiscriminately can be risky because secondary contexts occur with irrelevant events too. For example, an arrest can follow a theft instead of a terrorism event. This is why most current event extraction systems generally ignore the extractions in secondary contexts. Even within the main event description, a sentence may not appear to be relevant when viewed in isolation. For example, "He used a gun". Is the "gun" a weapon used in a terrorism event? Depending on the surrounding story context, such a sentence can be seen in the description of a terrorism event, a military operation event, or a common crime; accordingly, "He" may refer to a terrorist, a soldier, or a burglar. However, if we know that the larger context is discussing a relevant event, then we will be able to extract relevant event information from these contexts and improve the coverage of event extraction systems.

Second, with access to wider context, the accuracy of current event extraction systems can be improved too. Current event extraction systems will extract information if the local context contains seemingly relevant event keywords or phrases. However, depending on the larger context, they may not be referring to a relevant event due to ambiguity and metaphor. For example, "Obama was attacked" may lead to Obama being extracted as the victim of a physical attack, even if the preceding sentences describe a presidential debate

and the verb "attacked" is being used metaphorically.

Both of these problems tell us that it is necessary to develop better performing event extraction systems by modeling the influences of discourse during event extraction. In this dissertation, I will describe two discourse-oriented event extraction architectures that incorporate discourse information into event extraction to improve both extraction coverage and accuracy. In the following two subsections, I will briefly describe the design of these two models.

### 1.1.1 TIER: A Multilayered Event Extraction Architecture

The first one, called *TIER* [44], is a multilayered event extraction architecture that performs document level, sentence level, and noun phrase level text analysis to progressively "zoom in" on relevant event information. *TIER* represents a two-pronged strategy for event extraction that handles *event narrative* documents differently from other documents. I define an event narrative as an article whose main purpose is to report the details of an event. In contrast, I will refer to the documents that mention a relevant event somewhere briefly as fleeting references. I search for role fillers only in secondary contexts that occur in event narratives.

The main idea of *TIER* is to analyze documents at multiple levels of granularity in order to identify role fillers that occur in different types of contexts. My event extraction model (as shown in Figure 1.1) 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. At the top level, I train a document classifier to identify event narratives. At the middle level, I create two types of sentence classifiers. *Event sentence classifiers* identify sentences that mention a

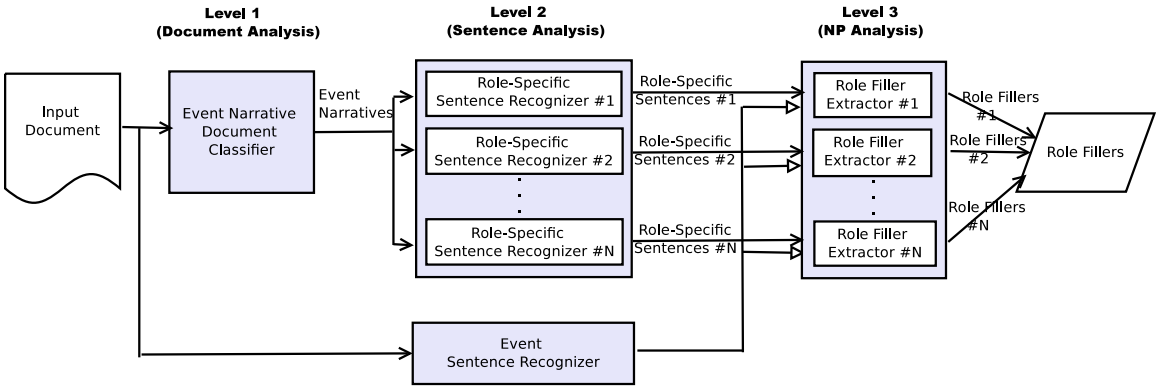


**Figure 1.1**: TIER: A Multilayered Architecture for Event Extraction

relevant event, and *role-specific sentence classifiers* identify sentences that contain possible role fillers irrespective of whether an event is mentioned. At the lowest level, I 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. 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 in event narratives. Identifying these *role-specific contexts* can root out important facts that would have been otherwise missed.

### 1.1.2 LINKER: A Bottom-up Event Extraction Architecture

The second model, called *LINKER* [45] (as illustrated in Figure 1.2), is a unified discourse-oriented event extraction architecture. In addition to a set of local role filler extractors as normally seen in event extraction systems, *LINKER* uses a single sequentially structured sentence classifier to explicitly model the contextual influences across sentences and identify event-related story contexts. The structured learning algorithm, conditional random fields (CRFs), explicitly models whether the previous sentence is an event context,

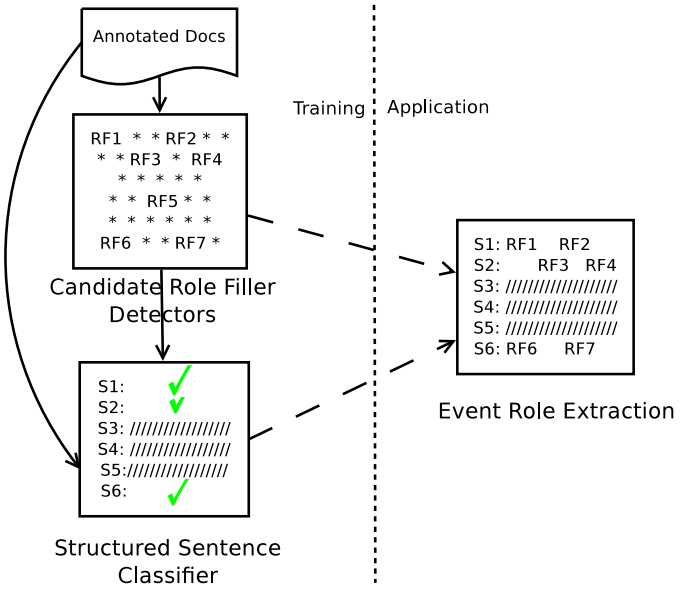


**Figure 1.2**: LINKER: A Bottom-up Architecture for Event Extraction

which captures discourse continuity across sentences. Furthermore, the structured sentence classifier can model a variety of discourse information as textual cohesion properties across sentences. Features are designed to capture lexical word associations, e.g., it is common to see "bombed" in one sentence and "killed" in the next sentence because bombing event descriptions are often followed by casualty reports. Features are also designed to capture discourse relations across sentences, e.g., if two sentences are in a causal relation, then probably both are event relevant sentences or neither of them is. In addition, its bottom-up design allows distributional properties of the candidate role fillers within and across sentences to be modeled as features. Intuitively, the presence of multiple role fillers within a sentence or in the preceding sentence is a strong indication that a relevant event is being discussed.

In *LINKER*, the sentence classifier sequentially reads a story and determines which sentences contain event information based on both the local and preceding contexts. Then, the structured sentence classifier and the set of local role filler extractors are combined by extracting only the candidate role fillers that occur in sentences that represent event contexts, as determined by the sentence classifier.

## 1.2 Multifaceted Event Recognition

Before giving documents to sophisticated event extraction systems, we want to ask if the documents actually contain any relevant events. Therefore, I also study event recognition that aims to identify documents describing a specific type of event. Accurate event recognition will improve event extraction accuracy because any extractions from documents that do not contain a relevant event will be false. Furthermore, event recognition is essential to many other event oriented applications. For example, with accurate event recognition, we can detect the first occurrences and the following mentions of a particular type of event; thus, we can track the dynamics of events over time.

Event recognition is a highly challenging task due to the high complexity and variety of event descriptions. It is tempting to assume that event keywords are sufficient to identify documents that discuss instances of an event. But event words are rarely reliable on their own. For example, consider the challenge of finding documents about civil unrest. The words *"strike"*, *"rally"*, and *"riot"* refer to common types of civil unrest, but they frequently refer to other things as well. A strike can refer to a military event or a sporting event (e.g., *"air strike"*, *"bowling strike"*), a rally can be a race or a spirited exchange (e.g., *"car rally"*, *"tennis rally"*), and a riot can refer to something funny (e.g., *"she's a riot"*). Event

keywords also appear in general discussions that do not mention a specific event (e.g., *"37 states prohibit teacher strikes"* or *"The fine for inciting a riot is $1,000"*). Furthermore, many relevant documents are not easy to recognize because events can be described with complex expressions that do not include event keywords. For example, *"took to the streets"*, *"walked off their jobs"*, and *"stormed parliament"* often describe civil unrest.

I propose multifaceted event recognition [46] to accurately recognize event descriptions in text by identifying event expressions as well as event facets, which are defining characteristics of the event. Event facets are essential to distinguish one type of event from another. For example, given the event expression "hit the village", depending on the agents, it might refer to a natural disaster event if the agent is "The flooding", or it might be describing an air strike if the agent is "The military bombs". Given the event expression "attacked", depending on "who" were "attacked" as the patient, it can be associated with a terrorism event ("civilians") or a general military operation ("soldiers"). Furthermore, event facets are so powerful that frequently, events can be recognized by only seeing multiple types of event facet information, without any event expression detected. For example, to identify documents describing civil unrest events, we feel confident to claim that a document is relevant if we pinpoint both the agent term "coal miners" and the purpose phrase "press for higher wages", even without detecting any event keyword such as "rally" and "strike".

The third component of my research is a bootstrapping framework that automatically learns event expressions as well as essential facets of events. The learning algorithm relies on limited supervision, specifically, a handful of event keywords that are used to create a pseudo domain-specific text collection from a broad-coverage corpus, and several seed terms for each facet to be learned. The learning algorithm exploits the observation that event expressions and event facet information often appear together in localized text regions that introduce an event. Furthermore, seeing more than one piece of event information together tends to validate that the text region is describing a relevant event and implies that additional event information may also be found in the same region. Therefore, in the first step, I identify probable event regions that contain multiple types of event facet information and extract event expressions based on dependency relations with event facet phrases. The harvested event expressions are added to an event phrase dictionary. In the second step, new phrases of an event facet are extracted from text regions containing an event phrase and phrases of the other event facets. The newly harvested event facet phrases are added to event facet dictionaries. The bootstrapping algorithm ricochets back and forth, alternately learning new event phrases and learning new event facet phrases, in an iterative process.

For example, civil unrest events are generally initiated by certain population groups, e.g., "employees", and with certain purpose, e.g., "demanding for better working conditions". Therefore, I identify agents and purposes as two facets of civil unrest events. To learn event expressions and event facet phrases, Figure 1.3 illustrates how the bootstrapping algorithm works.

## 1.3 Claims and Contributions

The primary contributions of this research are as follows:

1 *Both event extraction coverage and accuracy can be improved by incorporating discourse information across sentences to recognize event contexts before applying local extraction models.*

Event story telling generally spans over a text discourse. Accordingly, automatic event extraction systems should be able to model discourse phenomena to accurately locate pieces of event information in text. However, current event extraction systems generally process one sentence at a time and make extraction decisions relying on event clues from a limited text span as within the sentence. Due to lacking a global view of text contents, the current event extraction systems suffer from insufficient coverage and accuracy. In this research, I focus on improving extraction performance by incorporating discourse information across sentences to recognize event contexts before applying local extraction models. First, I designed a multilayered event extraction model, called *TIER*, to seek out event information that appears in secondary event contexts. The main idea of *TIER* is to zoom in on relevant event information, by using a document classifier and two types of sentence classifiers to analyze text at multiple

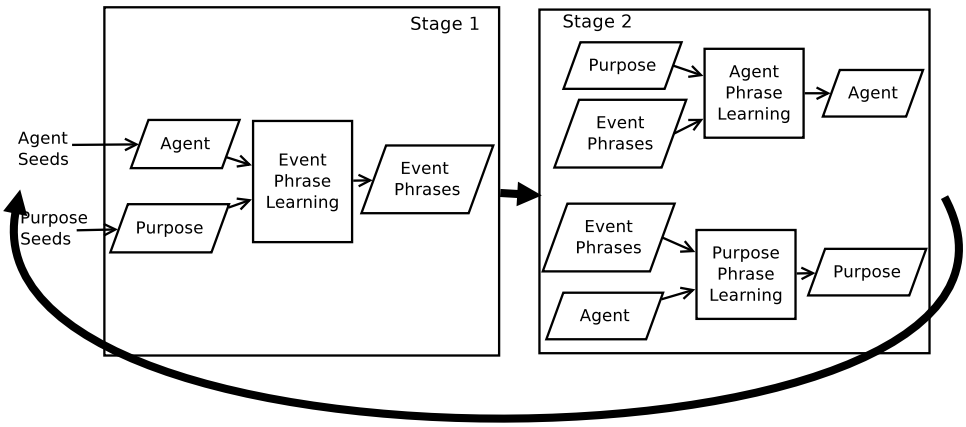


**Figure 1.3**: Bootstrapped Learning of Event Dictionaries

granularities. Later, I designed a unified discourse-guided event extraction architecture, *LINKER*, that explicitly models textual cohesion properties across sentences to accurately find out event-related contexts, using a single structured sentence classifier. Evaluation on two event domains shows that my discourse guided event extraction architectures have improved both event extraction coverage and accuracy.

2 *Event defining characteristics (event facets), in addition to event expressions, can be used to accurately identify documents describing a particular type of event.*

Finding documents that describe a specific type of event is a challenging task due to the high complexity and variety of event descriptions. Event keywords tend to be ambiguous and are not sufficient to identify documents that discuss event instances of a particular type. I propose multifaceted event recognition to accurately recognize event descriptions in text by identifying event expressions as well as event facets, which are defining characteristics of the event. Event facets, such as agents, purpose, and effects of events, are essential to distinguish one type of event from another. I also propose a bootstrapping framework to automatically learn event expressions as well as essential facets of events, requiring only unannotated text and minimal human supervision. Evaluation on two event domains shows that multifaceted event recognition can yield high accuracy.

## 1.4 Navigating This Dissertation

The rest of this dissertation is summarized as follows.

- Chapter 2 describes existing work in event extraction and recognition studies. This chapter explains the limitations of current event extraction and recognition systems and demonstrates how the research presented in this dissertation contributes in addressing these limitations.

- Chapter 3 presents the details in designing *TIER*, the multilayered event extraction architecture, which can seek out event information from seconday event contexts. To motivate, I will first discuss secondary contexts, in contrast with primary event contexts. Then, I will demonstrate two types of documents that mention relevant events, event narratives, and fleeting references, with respect to how event information was conveyed in text. Then, this chapter presents design details of the four components that constitute the multilayered event extraction architecture.

- Chapter 4 presents the details in designing the unified discourse-guided event extraction architecture, *LINKER*. This chapter illustrates the bottom-up system design and discusses the main idea that uses a single structured event context recognizer to identify all the event-related sentences in a document. After that, this chapter elaborates the linguistic discourse features that are used in the structured event context recognizer to capture textual cohesion properties aross sentences.

- Chapter 5 demonstrates the multifaceted event recognition approach. This chapter discusses the insufficiency of using event keywords for event recognition and illustrates how event defining characteristics (facets) can be helpful to recognize events of a particular type in text. This chapter includes a thorough discussion of event facets in a variety of events. Then, this chapter describes details of the bootstrapping framework that is effective in acquiring both event expression and event facet information from unannotated text. In the evaluation section, I also examines whether multifaceted event recognition can be used to improve event extraction performance, especially with respect to extraction accuracy.

- Chapter 6 discusses the conclusions that we can draw from the dissertation. Following the conclusions, this chapter suggests the future directions that can potentially lead to further progress in event-oriented information extraction research.

# CHAPTER 2

# BACKGROUND AND RELATED WORK

Information Extraction (IE) ([42]) is a major application area of Natural Language Processing (NLP). Among various subdisciplines in Information Extraction, Event Extraction (i.e., recognizing and extracting events from texts) has attracted intensive research attention over the last decades (e.g., [8, 94, 95, 47, 117, 33]) and continues to thrive in recent years (e.g., [23, 18, 30, 106, 69, 102, 100]). This dissertation focuses on improving event extraction performance by exploring discourse-guided approaches and incorporating accurate event recognition.

In the following sections, I will first introduce the event extraction task and discuss different "genres" of event extraction research, then I will briefly mention standard evaluation datasets that are available for event extraction research. Next, I will talk about two streams of classic approaches that have been developed for event extraction. Then, I will focus on discussing recent advances in event extraction that are closely related to my research as presented in this dissertation. I will also compare event extraction methods with the approaches that are developed for several other related NLP tasks. Finally, I will cover various types of research work that are related to recognizing events in texts (i.e., event recognition).

## 2.1 Event Extraction Task and Datasets

There have been several community-wide performance evaluations dedicated to advancing event extraction research. These evaluations have shaped event extraction as a major research area of natural language processing and significantly influenced event extraction research by revealing a diverse set of extraction approaches and providing standard annotated datasets for evaluating the future event extraction systems.

### 2.1.1 Message Understanding Conferences

Among these efforts, there was a series of Message Understanding Conferences (MUC-1 to MUC-7), spanning over a decade (from 1987 to 1997), that defined the template-based event extraction task and attracted a significant amount of attention from the research

community. In template-based event extraction, the goal of event extraction systems is to identify and extract pieces of key event information in texts and classify them into their corresponding event roles. Event roles can specify the participants of events, objects that are involved in events, or properties associated with an event. The extracted text snippets that fill certain event roles are called *event role fillers*, which are generally noun phrases. Template-based event extraction also requires template generation specifying each event with its set of role fillers, which is complex because many documents have multiple templates (i.e., they discuss multiple events).

Multiple event extraction evaluation datasets were created in the MUCs. The annotated datasets are mainly unstructured texts, military reports, or news reports, and each dataset was created for a specific domain. The event domains vary from terrorism events [76], corporate joint ventures [77] and management successions [78], to airplane crashes [79]. The number of "string-fill" event roles varies too. Several event roles were defined for terrorism events, including perpetrators, victims, physical targets, and weapons, while a smaller number of event roles were defined for events such as airplane crashes or joint ventures.

Many of these datasets have become benchmark collections for evaluating event extraction systems. Events of a particular type are sparse in a general news stream, so the MUCs mimic a realistic event extraction task where the IE system must determine whether a relevant event is present in the document before extracting role fillers. Consequently, most of the Message Understanding Conference data sets contain (roughly) a 50/50 mix of relevant and irrelevant documents (e.g., MUC-3, MUC-4, MUC-6, and MUC-7 [41]).

### 2.1.2 Automatic Content Extraction

Automatic Content Extraction (ACE) [1] is another research endeavor (1999 - 2008) that focuses on developing information extraction techniques from unstructued texts. ACE presents several challenges to participants, including identifying entity mentions, classifying semantic relations between pairs of entity mentions, and extracting events in texts. One characteristic of ACE is that evaluation datasets were provided in multiple languages. In addition to the English language, ACE (e.g., ACE 2005, ACE 2007, ACE 2008) provided evaluation data in other languages too, including Arabic, Chinese and Spanish. Therefore, ACE has successfully stimulated wide information extraction research interests across many countries.

In contrast to the MUCs, ACE defined a rich set of event types and the events annotated in ACE data sets are not with respect to a particular domain. Instead, multiple types of

events can be annotated in one single document. Furthermore, ACE systems are designed to process general news articles and extract general events, such as interaction, movement, transfer, creation, and destruction events. Furthermore, as written in guidelines for both annotation and evaluation purposes, in addition to event arguments and attributes, each event mention must have an anchor or trigger word associated with it.

### 2.1.3 Other Datasets for Event Extraction Research

Several other event extraction data sets have been created, mostly by individual research groups. Some well-known ones include the data set for the domain of corporate acquisitions [33, 34, 30], job postings [18, 34], and seminar announcements [33, 24, 23, 30, 37]. Different from the MUC and ACE data sets, which mainly consist of unstructured texts, documents in some of the data sets, specifically job postings and seminar announcements, are semistructured. For example, job postings generally put the post date and job title at the beginning of a post. There are also more recent data sets established to facilitate event extraction research, including the disease outbreak data set [85] and several biomedical event extraction data sets (e.g., [72]). The disease outbreak data set contains documents that are collected from an open-source, global electronic reporting system for outbreaks of infectious diseases, ProMed [91]. The biomedical data set has been used in the BioNLP09 [58] shared task, which focuses on the extraction of biomolecular events.

### 2.1.4 Research in This Dissertation

My dissertation focuses on extracting events from free texts as in the MUC evaluations. However, while the complete event extraction task involves template generation, my work focuses on extracting individual facts and not on template generation per se (e.g., I do not perform coreference resolution or event tracking). As noted earlier, most MUC data sets contain a mix of relevant and irrelevant documents and represent a more realistic setting for the event extraction task. In addition, compared to the event extraction task in ACE, the MUC evaluations target a particular type of event. Among a series of MUC data sets, MUC-4 terrorism corpus [76] is a standard benchmark collection for evaluating event extraction systems and is especially interesting because it defines a rich set of event roles in a terrorism event template. Figure 2.1 shows a sample document in MUC-4 corpus and Figure 2.2 shows its associated event template with the defined event roles filled.

In this dissertation, I propose new event extraction architectures that improve both event extraction coverage and accuracy by incorporating discourse information across sentences to recognize event contexts before applying local extraction models. I will evaluate my new

BOGOTA, 13 FEB 90 (RADIO CADENA NACIONAL) -- [TEXT] THE STATE'S SECRET SERVICES STILL HAVE NO CLUES REGARDING THE TWO U.S. CITIZENS WHO WERE KIDNAPPED IN THE PAST FEW HOURS BY GUERRILLAS OF THE ELN [ARMY OF NATIONAL LIBERATION] DURING OPERATIONS IN MEDELLIN.

THE VICTIMS WERE IDENTIFIED AS DAVID LECKY, DIRECTOR OF THE COLUMBUS SCHOOL, AND JAMES ARTHUR DONNELLY.

BOTH WERE KIDNAPPED BY THE SO-CALLED NELSON MANDELA CELL OF THE ELN 48 HOURS BEFORE THE PRESIDENTIAL DRUG SUMMIT IN CARTAGENA.

MADRID EFE IN SPANISH AT 2132 GMT ON 13 DECEMBER REPORTS THAT BOGOTA RADIO CADENA NACIONAL LISTS THE KIDNAP VICTIMS AS "LESLIE KENT, PROFESSOR OF THE COLUMBUS SCHOOL IN MEDELLIN, AND JAMES ARTHUR DONNELLY..."

**Figure 2.1**: A Sample Document from the MUC-4 Corpus (ID: TST2-MUC4-0039)

event extraction architectures using the MUC-4 terrorism data set and a new data set on civil unrest events, created in a similar style as the MUC data sets (see Section 3.5.1 for more details), to show the generality of my proposed approaches. I will also use the same two data sets to evaluate the effectiveness of my multifaceted event recognition to improve the accuracy of event extraction systems.

## 2.2 Classic Approaches for Event Extraction

Contexts around a potential extraction play an important role in determining its event role. For example, in terrorism events, a person can be a perpetrator or a victim depending on different contexts. Most event extraction systems scan a text and search in small context windows using patterns or a classifier. Pattern-based approaches (e.g., [8, 57, 94]) extract event role fillers by matching linguistic patterns with the local context of text segments that have been identified as potential extractions. Therefore, the extraction performance greatly depends on the quality of the used linguistic patterns. In Section 2.2.1, I will discuss different methods that are used to derive extraction patterns. In contrast, classification-based approaches generally train statistical classifiers to identify event role fillers. These approaches can easily leverage a variety of contextual clues and make extraction decisions based on statistical properties of a potential extraction being an event role filler. In recent years, classifier approaches have been frequently applied for extracting information from free texts.

### 2.2.1 Pattern-based Approaches

Patterns are derived from texts that contain event role fillers and capture lexical, syntactic, or semantic properties that are commonly associated with a particular type

| | |
|---|---|
| 0. MESSAGE: ID | TST2-MUC4-0039 |
| 1. MESSAGE: TEMPLATE | 1 |
| 2. INCIDENT: DATE | (13 FEB 90) / (12 FEB 90 - 13 FEB 90) |
| 3. INCIDENT: LOCATION | COLOMBIA: MEDELLIN (CITY) |
| 4. INCIDENT: TYPE | KIDNAPPING |
| 5. INCIDENT: STAGE OF EXECUTION | ACCOMPLISHED |
| 6. INCIDENT: INSTRUMENT ID | * |
| 7. INCIDENT: INSTRUMENT TYPE | * |
| 8. PERP: INCIDENT CATEGORY | TERRORIST ACT |
| 9. PERP: INDIVIDUAL ID | "NELSON MANDELA CELL OF THE ELN" / |
| "NELSON MANDELA CELL" | |
| 10. PERP: ORGANIZATION ID | "ARMY OF NATIONAL LIBERATION" / "ELN" |
| 11. PERP: ORGANIZATION CONFIDENCE | REPORTED AS FACT: "ARMY OF NATIONAL |
| LIBERATION" / "ELN" | |
| 12. PHYS TGT: ID | * |
| 13. PHYS TGT: TYPE | * |
| 14. PHYS TGT: NUMBER | * |
| 15. PHYS TGT: FOREIGN NATION | * |
| 16. PHYS TGT: EFFECT OF INCIDENT | * |
| 17. PHYS TGT: TOTAL NUMBER | * |
| 18. HUM TGT: NAME | "DAVID LECKY" |
| | "JAMES ARTHUR DONNELLY" |
| 19. HUM TGT: DESCRIPTION | "DIRECTOR OF THE COLUMBUS SCHOOL" |
| / "DIRECTOR": "DAVID LECKY" | |
| 20. HUM TGT: TYPE | CIVILIAN: "DAVID LECKY" |
| | CIVILIAN: "JAMES ARTHUR DONNELLY" |
| 21. HUM TGT: NUMBER | 1: "DAVID LECKY" |
| | 1: "JAMES ARTHUR DONNELLY" |
| 22. HUM TGT: FOREIGN NATION | UNITED STATES: "JAMES ARTHUR |
| DONNELLY" | |
| | UNITED STATES: "DAVID LECKY" |
| 23. HUM TGT: EFFECT OF INCIDENT | - |
| 24. HUM TGT: TOTAL NUMBER | 2 |

**Figure 2.2**: Annotated Event Template for the Sample Document

of event role. Early event extraction systems used hand-crafted patterns (e.g., [8, 61]). FASTUS ([8]) extracted information using patterns that are encoded as finite state machines operating on phrasal decompositions of a sentence. In FASTUS, 95 hand-crafted patterns were used to extract event information from the MUC-4 terrorism data set. For example, one pattern used to identify victim role fillers is "killing of <VICTIM>". The experiments show that the pattern-based approach can extract information from texts effectively and efficiently.

However, creating patterns manually for each event domain is tedious and time consuming, so more recent systems generate patterns or rules automatically using supervised learning (e.g., [57, 94, 107, 47, 33, 24, 18]). Supervised learning algorithms use human annotated event information as supervision when inducing linguistic patterns. PALKA ([57]) acquires domain-dependent semantic patterns corresponding to predefined frame representations. PALKA is semi-automatic because in the pattern acquisition process, it needs simple forms of human interaction to determine the relevancy of a clause and

a relevant phrase in the clause, with respect to frame definitions. AutoSlog ([94]) automatically generates domain-specific extraction patterns using a training corpus tagged with the targeted event information. CRYSTAL [107] automatically induces generalized linguistic patterns ("concept-node definitions") by locating and comparing definitions that are highly similar and creating unified definitions. Sudo et al. [111] discuss the limitations of prior extraction pattern representations and introduce the subtree extraction model that is based on arbitrary subtrees of dependency trees and can extract entities beyond direct predicate-argument relations.

Relational learning algorithms [33, 18] have been shown effective to induce extraction rules or patterns. These learning algorithms vary in how how they induce patterns from texts. SRV [33] induces rules from general to specific (top-down). Specifically, SRV starts with all negative examples and any positive examples not covered by already induced rules (all positive examples at the beginning), and adds predicates greedily to cover as many positive and as few negative examples as possible. Predicates are formed using simple token-based features and relational features. Instead, RAPIER [18] is a bottom-up learning algorithm that consists primarily of a specific to general search. It creates very specific rules and then generalizes those to cover additional positive examples. Specifically, RAPIER directly induces extraction patterns that each is formed by three parts: a prefiller pattern that must match the text immediately preceding the slot-filler, a slot-filler pattern that must match the actual slot-filler, and a postfiller pattern that must match the text immediately following the filler. RAPIER constructs pattern rules for each training instance, then takes random pairs of rules, generalizes each pair, and selects the best generalization as the new rule.

As mentioned above, supervised pattern learning algorithms require annotated event information to learn patterns. However, event information is expensive to annotate. To further reduce human supervision needed to learn patterns, several weakly supervised learning approaches were proposed (e.g., [95, 96, 117, 111, 110]). AutoSlog-TS ([95]) eliminated the dependency on an annotated training corpus and only uses untagged text. Specifically, AutoSlog-TS is built on top of AutoSlog ([94]), which was adapted to exhaustively generate patterns that can be used to extract all noun phrases in texts. Then, AutoSlog-TS learns good extraction patterns by ranking patterns based on the statistical probabilities that patterns occur in event-relevant documents. Later, Riloff and Jones [96] presents a multilevel bootstrapping algorithm that generates both extraction patterns and a semantic lexicon simultaneously in a iterative process. In this algorithm, only a handful

of seed words for each semantic category is used as human supervision. Specifically, a mutual bootstrapping technique is used to alternately select the best extraction patterns and bootstrap its extractions into the semantic lexicon, which is used to select the next extraction pattern. Similarly, Snowball [2] requires only a handful of training examples from users. The initial examples are used to generate extraction patterns, which in turn are used to extract new training instances from the document collection. ExDISCO ([117]) reduces the required supervision to several initial "seed" patterns in an iterative learning process. ExDISCO uses the initial patterns to find the first batch of event-relevant documents, from which, more patterns are learned. Then, the learned patterns are used to retrieve more event-relevant documents. The learning process iterates.

In recent years, there have also been learning algorithms that proceed in an unsupervised manner (e.g., [22, 106, 105]). Chambers and Jurafsky ([22]) acquire event words from an external resource, WordNet [74], group the event words to form event scenarios, and group extraction patterns for different event roles. Shinyama and Sekine ([106]) proposed preemptive information extraction that attempts to automatically create feasible IE systems in advance without human intervention. Mainly, they cluster a set of articles from the web that essentially describe a particular type of event and discover patterns that can extract entities playing the same role.

### 2.2.2 Classifier-based Approaches

Many classifiers have been created to label phrases or single tokens with respect to an event role (e.g., [32, 23, 30, 63, 120]). Freitag ([32]) suggested to use three types of learners, rote memorization, term-space text classification, and relational rule induction, to examine potential extractions, and make extraction decisions using combined probabilities that are mapped from individual learners' confidences. Chieu and Ng ([23]) proposed to apply a specific machine learning algorithm, maximum entropy, to weigh multiple sources of extraction evidence in a single statistical model. Their extraction evidences are largely derived from the local contexts of target phrases and the phrases themselves. A rich set of specific features were used to train their models. Note that in this work, Chieu and Ng also learned to build event templates using the product of entity pair relation probabilities. In addition, Wrap-Up [108] is a trainable IE discourse component that learns to construct event templates.

Instead of classifying phrases with respect to an event role ([32, 23]), methods ([30, 63, 120]) have also been proposed to classify single tokens to indicate if each token is part of an extraction or not. Finn and Kushmerick ([30]) treat the identification of extractions'

start and end positions as distinct token classfication tasks and train separate statistical classifiers for each. Specifically, they chose support vector machines as their machine learning algorithm. The features that are used to classify each token include the token itself, part-of-speech, chunking, orthographic, and gazetteer information. Later, instead of using only local features, Finkel et al. ([29]) used long distance dependency models to enforce label consistency and extraction template consistency constraints. In their work, Gibbs sampling was employed to perform approximate inference which runs in tractable time. Li et al. ([63]) also applied machine learning algorithms to classify each token in texts. They especially emphasized the importance of using an uneven margins parameter in support vector machines and perceptrons to tackle the skewed distributions between positive and negative instances, which is notable in event extraction task because relevant event information is only sparsely scattered in texts. Yu et al. ([120]) proposed a cascaded event extraction approach and showed that it is effective to automatically extract information from resumes. Their approach first identifies blocks of texts that have labels (e.g., *Personal Information* in their case.), then classify each token (mainly punctuations because potential extractions are generally separated by punctuations in their case) within certain types of blocks with respect to a specific type of information (e.g., applicants' names).

Recently, structured sequence tagging models ([93, 71, 59]), especially Conditional Random Fields ([59, 86]) and their variants or generalized models (e.g., [15]), have proved to be effective for information extraction tasks. Instead of labeling an individual phrase or a single token independently, structured sequence tagging models consider mutual dependencies between labels that are assigned to neighboring text units, and label a sequence of tokens. Among these, Lu and Roth ([66]) uses the latent-variable semi-Markov conditional random fields for jointly extracting event role fillers from texts.

## 2.3 Recent Advances in Event Extraction Research

Most of the classic approaches heavily rely on the local context of individual potential extractions when making decisions. However, recent work has begun to leverage additional contextual information and consider associations among candidate role fillers to improve extraction performance.

### 2.3.1 Using Event Clues from Sentential Contexts

Research has been conducted to explore sentential contexts ([69, 37, 85, 102]) when identifying individual role fillers. Maslennikov and Chua ([69]) propose to view event fact extraction at the multiresolution layers of phrases, clauses, and sentences using dependency

and discourse relations. Specifically, they use both discourse trees and local syntactic dependencies *within* sentences in a pattern-based framework. Their extraction framework, called ARE (short for Anchor and Relation), uses clause-level discourse relations to both filter noisy dependency paths and to increase reliability of dependency path extraction. ARE starts with extracting candidate entities (anchors) of appropriate anchor types, evaluates the relationships between them, further evaluates all possible candidate templates, and outputs the final template.

Some research work ([37, 85, 102]) has trained separate sentence classifiers to identify event-relevant sentences and then consider extracting event information mainly from the relevant sentences as identified by the sentence classifiers. Gu and Cercone ([37]) introduce the concept of extraction redundancy that many current sequential labeling IE systems often produce undesired redundancy extractions. To address this issue, they propose a segment-based two-step extraction approach in which a segment retrieval step is imposed before the extraction step. Specifically, they created HMMs to first identify relevant sentences and then trained another set of HMMs to extract individual role fillers from the relevant sentences. Patwardhan and Riloff ([85]) distinguish primary and secondary extraction patterns and argue that primary extraction patterns can be used by themselves to extract event information while the use of secondary patterns should be constrained within event-relevant sentences. They also designed a weakly-supervised learning paradigm to learn to recognize event sentences. Later, Patwardhan and Riloff ([102]) also proposed a unified model for event extraction, called GLACIER, that jointly considered sentential evidence and local phrasal evidence in a probabilistic framework when extracting role fillers. GLACIER uses sentence classifiers that were trained with supervised learning. Experimental results show that GLACIER balanced the influence of sentential context with local contextual evidence and improved the performance of event extraction.

Overall, by looking beyond the immediate contexts of potential extractions, previous models have achieved better extraction performance. However, none of these systems explored contexts out of the sentence containing the candidate role fillers. In contrast, my discourse-guided event extraction models explore how an event is described in a document and explicitly model the contextual influences across a document, including lexical cohesion properties, discouse relations, and domain-specific candidate role filler distributions.

### 2.3.2 Leveraging Associations across Event Role Fillers

There has been research ([64, 48, 66]) that mines associations among candidate role fillers to improve extraction performance. One advantage of such research is that this approach

can easily go beyond the local context of an individual candidate role filler and leverage information about other role fillers from the same sentence, the same document, or even across documents to make better extraction decisions. Liao and Grishman ([64]) observed that correlations exist between occurrences of different types of events and different event arugments. For example, they found that Attack, Die, and Injure events often occur together and Victims of a Die event are frequently the Targets of an Attack event. Following this observation, they introduced cross-event information to enhance the performance of multi-event-type extraction by using information about other types of events to make predictions or resolve ambiguities regarding a given event. Specifically, they calculated document-level role filler statistics and used the co-occurrence information between different types of events and between different role fillers as features to train better extraction models. Ji and Grishman [48] noted that many events are reported multiple times, in different forms, both within the same document and within topically related documents. Therefore, they proposed to take advantage of alternate descriptions of the same event fact and propagate consistent event arguments across sentences and documents.

More recently, Lu and Roth [66] use the latent-variable semi-Markov conditional random fields to encode role filler dependencies (e.g., as shown in their paper, an AGENT and an VICTIM are often seen with "killed" in between) as structured preferences in a model learning process. Therefore, this approach enables joint extraction of event role fillers from texts. Li et al. [62] uses structured perceptron to jointly predict event triggers and their arguments within a sentence. Various global features are designed to model dependencies between two triggers and among multiple arguments.

Overall, these models focus on the interrelations between different role fillers or different mentions of the same role filler in a sentence ([66, 62]), document ([64]), or corpus ([48]) and use the leveraged role filler associations to aid event role classification. However, different from my research of discourse-guided event extraction, neither of them concentrates on exploring wider contexts across sentences other than role fillers associations, and these contexts include lexical links and discourse relations across sentences.

## 2.4 Other Related NLP Research Areas

Event extraction is closely related to several other NLP areas, such as relation extraction and semantic role labeling, but these tasks each have a unique goal and present different challenges to computational linguists. In the following subsections, I briefly compare event extraction research with several other related NLP study areas. In addition, I will also

discuss research in text segmentation and modeling document-level content transitions, which are closely related to my research on discourse-guided event extraction architectures.

### 2.4.1 Relation Extraction

Research has been done on relation extraction (e.g., [101, 121, 80, 17, 16]), which aims to identify predefined semantic relations between pairs of entities in text. In contrast, an event can consist of more than two entities. However, as discussed earlier, many classic event extraction methods decompose event extraction to extracting one event role filler a time and thus, the event extraction task can be viewed as classifying the relation between an event trigger and a potiental extraction.

Relation extraction methods mainly fall into two categories, feature-based methods and kernel-based methods. Feature-based methods ([80, 125]) for relation extraction encode various lexical, syntactic, and semantic features explicitly when training classification models. In comparison, kernel-based methods ([121, 27, 17, 122, 123, 126]) explore the parsing or dependency structural information of the text between two entities implicitly by computing the similarity between two objects via a kernel function.

Recently, Bunescu and Mooney [16] proposed a weakly supervised relation extraction approach that requires only a few pairs of well-known entities, where some (positive) pairs clearly exhibit a particular relation while others (negative) do not. Sentences containing the examples are extracted from the web. They assume that many sentences containing a positive entity pair state the desired relation, and none of the sentences containing a negative entity pair state the relation. Multiple instance learning was used to exploit this weakly supervised learning setting. Lately, researchers have used distant supervision ([75, 118, 43]) leveraged from existing databases to initiate the learning of relation extractors with many more entity pairs.

### 2.4.2 Open Information Extraction

Open Information Extraction (Open IE) is the task of extracting assertions from massive corpora, commonly the web, without requiring a prespecified vocabulary. Open IE techniques (e.g., KNOWITALL [28] and TEXTRUNNER [9]) have been developed to generate a large set of domain-independent relational tuples from texts in the web. Each of the learned relational tuples generally consists of a pair of entities and a string to represent the relation between the entity pair. NELL (short for Never Ending Language Learning, [19, 20]) is another IE learner that is initiated by a handful of relation pairs and continues to accumulate learned relations. NELL simultaneously learns classifiers for different entity

categories and relations aided by an ontology defining constraints that couple the classifier training.

### 2.4.3 Semantic Role Labeling

A large amount of research has been conducted for semantic role labeling ([35, 116, 112, 92, 38, 119, 21]), also called shallow semantic parsing, which aims to detect semantic arguments of a predicate and classify the arguments into their semantic roles. The predicate is usually a verb or a noun in a sentence and the arguments are mostly from the same sentence as the predicate. Compared to event extraction, semantic role labeling focuses on semantic analysis of individual predicates, instead of extracting certain types of information with respect to an event. Frequently, fillers of a certain type of event role can perform distinct semantic roles when associated with different predicates. For example, in terrorism events, perpetrators can be both the agent of actions such as "bombed", and the patient of predicates such as "arrested".

### 2.4.4 Text Segmentation

My event extraction research is loosely related to text segmentation ([39, 12, 56, 49, 67, 54]), which aims to divide a document into consecutive segments such that each segment describes a coherent central topic. Similarly, my research targets better event extraction performance by identifying contexts in a document that describe a particular type of event. However, text segmentation systems generally produce text segments that consist of a series of sentences discussing the same topic. In comparison, my discourse-guided event extaction architectures detect event contexts as fine as an individual sentence in a document.

### 2.4.5 Document-level Content Models

My work is also related to the document-level content models introduced by [10], which utilized a novel adaptation of the generative sequential model HMMs [93] to capture the topics that the texts address and the transitions between topics. The learned topic sequences improved two applications, information ordering and extractive summarization. Recently, [104] incorporates the latent content structure directly into two text analysis tasks, extractive summarization and sentiment analysis, in a joint learning framework. In one of my discourse-oriented event extraction architectures, I will include a structured sentence classifier to model the textual cohesion across sentences in a document. However, the structured sentence classifier as included in my second discourse-guided event extraction model is different from the structured content models, because the former is trained discriminatively

and with respect to one particular task.

## 2.5 Event Recognition

In this dissertation, I study event recognition too, which aims to identify documents describing a specific type of event. This is different from event extraction studies that aim to produce full representations of events. There has been relatively little work that focuses specifically on the event recognition task, but event recognition has been studied in the context of other tasks.

### 2.5.1 "Text Filtering" in Event Extraction

There has been a lot of research on event extraction (e.g., [1, 8, 95, 117, 23, 18, 111, 110, 105]), where the goal is to extract facts about events. The MUC-4 evaluation [76] included "text filtering" results that measured the performance of event extraction systems at identifying event-relevant documents. The best text filtering results were high (about 90% F score), but relied on hand-built event extraction systems. More recently, some research has incorporated event region detectors into event extraction systems to improve extraction performance [37, 85].

### 2.5.2 Event Detection in Social Media

There has been recent work on event detection from social media sources [11, 88]. Some research identifies specific types of events in tweets, such as earthquakes [103] and entertainment events [13]. There has been work on event trend detection [60, 70] and event prediction through social media, such as predicting elections [113, 26] or stock market indicators [124]. [100] generated a calendar of events mentioned on twitter. [73] proposed structured retrieval of historical event information over microblog archives by distilling high-quality event representations using a novel temporal query expansion technique.

### 2.5.3 Text Classification

Text classification techniques [81, 31, 52] categorize documents according to their topics or themes. There is also text classification research that has focused on event categories. [97] used an information extraction system to generate *relevancy signatures* that were indicative of different event types. This work originally relied on manually labeled patterns and a hand-crafted semantic dictionary. Later work [98] eliminated the need for the dictionary and labeled patterns, but still assumed the availabilty of relevant/irrelevant training texts, and required a parser to match the linguistic patterns in new texts.

### 2.5.4 Topic Detection and Tracking (TDT)

Event recognition is also related to Topic Detection and Tracking (TDT) [4, 3] which addresses event-based organization of a stream of news stories. Event recognition is similar to New Event Detection (NED), also called First Story Detection (FSD), which is considered the most difficult TDT task [5]. Typical approaches reduce documents to a set of features, either as a word vector [6] or a probability distribution [50], and compare the incoming stories to stories that appeared in the past by computing similarities between their feature representations. Recently, event paraphrases [87] have been explored to deal with the diversity of event descriptions. However, the NED task differs from our event recognition task because we want to find all stories describing a certain type of event, not just new events.

### 2.5.5 Faceted Search vs. Multifaceted Event Recognition

Faceted search ([40, 114]) enables users to explore a multidimensional information space by combining text search with a progressive narrowing of choices in each dimension. Information dimensions are also called facets, which correspond to properties of the information elements, e.g., webpages, and are useful to organize a large collection of information. However, in my multifaceted event recognition approach, facets refer to specific defining characteristics of events, e.g., purpose of events. Furthermore, I use facets, in addition to event expressions, to accurately identify events of a particular type.

## 2.6 Conclusions

In this chapter, I first overviewed the event extraction task and surveyed classic approaches for extracting events from texts. Then, I discussed recent advances in event extraction research and compared my work with the newly proposed approaches. I also briefly reviewed several NLP research areas that are related to event extraction. Finally, I discussed previous event recognition research that has been conducted mainly under other guises.

# CHAPTER 3

# *TIER*: A MULTILAYERED EXTRACTION ARCHITECTURE

As explained in Chapter 1, the goal of event extraction systems is to identify entities and objects (mainly noun phrases) that perform key roles in events. Most current event extraction systems heavily rely on local context when making extraction decisions. 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. Therefore, most current event extraction systems generally tackle event recognition and role filler extraction at the same time and primarily recognize contexts that explicitly refer to a relevant event.

However, lacking the view of wider context limits the coverage of traditional event extraction systems because 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. I will refer to these types of sentences as "secondary contexts" because they are generally not part of the main event description.

To illustrate how secondary event contexts occur in event descriptions, Figure 3.1 shows a typical terrorism event story. The news article starts with introducing a kidnapping event at the beginning of the story, followed by an elaboration on the victim names and affiliation information in the context of a person identification. Then, the article reverts back to convey more information about the kidnapping event, including specifically when it happened and the perpetrators involved. The mission of event extraction systems is to extract the underlined pieces of text and label each with their corresponding roles.

It is relatively easy for current event extraction systems to extract the event information in the first and third sentences because both sentences explicitly refer to the kidnapping event (primary contexts). However, the middle sentence, without consulting the wider

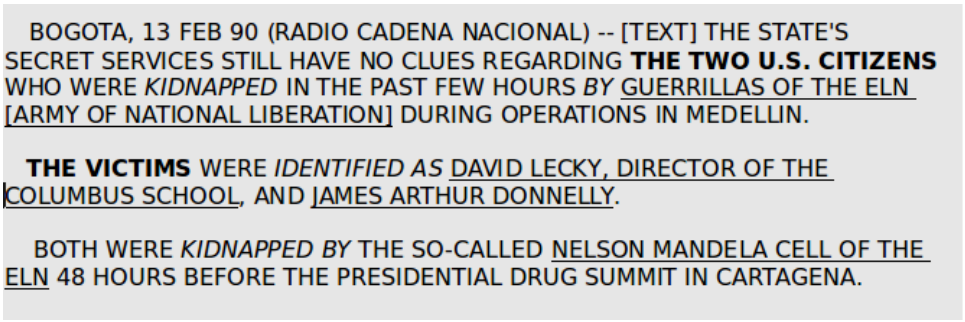


**Figure 3.1**: A Terrorism Event Story (ID: TST2-MUC4-0039, excerpted)

discourse, describes person identification that is not directly related to terrorism events. The victim information within it tends to be overlooked by current event extraction systems because this sentence does not contain any mention of the kidnapping event. The second sentence is a good representative of secondary contexts that commonly follow the main event descriptions and describe activities that tend to happen after the events of interest.

To extract event information buried in secondary contexts, one option is to carry out discourse analysis that can explicitly link secondary contexts to the main event, but discourse modeling by itself is a difficult problem. As in Figure 3.1, if we can accurately detect that the noun phrases "THE TWO U.S. CITIZENS" from the first sentence and "THE VICTIMS" from the second sentence actually refer to the same entity, then we can associate the person identification context with the kidnapping event and extract the victim information from the later sentence. However, entity coreference resolution across sentences is still a challenging problem in its own right.

In this chapter, I will introduce a multilayered event extraction architecture, *TIER*, that can effectively seek event information out of secondary contexts and therefore improve the extraction coverage, while maintaining high precision. In the following sections, I will first discuss the challenges and obstacles of identifying secondary event contexts. Then I will present my multilayered event extraction architecture that can extract event information from both primary and secondary contexts. I will focus on the main idea that is to analyze text in multiple granularities (document, sentence, and noun phrase levels) to zoom in on the relevant event information. I will also elaborate on the features and machine learning settings used to implement a working system of the multilayered event extraction architecture. Finally, I will present the evaluation results on two event domains. The first data set is a standard event extraction benchmark collection for terrorism events. The second data set was created recently to evaluate event extraction for civil unrest events.

## 3.1 Detecting Secondary Event Contexts

My goal here is to improve event extraction by learning to identify secondary role filler contexts in the absence of event keywords. I 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, my model should recognize that a sentence describing an arrest probably includes a reference to a perpetrator, even though the crime itself is reported elsewhere. Please refer to subsection 3.4.1 for the details of creating *role-specific* sentence classifiers.

Extracting information from these secondary contexts can be risky, however, unless we know that the larger context is discussing a relevant event. As an example, Figure 3.2 shows another terrorism event story. Unlike the one in Figure 3.1, this document focuses on an irrelevant topic about the presence of British and Israeli mercenaries and only briefly mentions terrorism events (mass murders) towards the end of the document. However, the person identification contexts, while exactly the same as in the story in Figure 3.1, do not contain victim information of terrorism events, because their surrounding larger contexts are mainly about an irrelevant topic. If event extraction systems scrutinize secondary contexts and extract their noun phrases indiscriminately, false hits will be produced that will affect extraction accuracy. Therefore, it is necessary to distinguish this type of documents from the ones as in Figure 3.1, to well identify valid secondary event contexts.

Specifically, I define an event narrative as an article whose main purpose is to report the details of an event. I will refer to the documents similar to the one in Figure 3.2 as *fleeting reference* texts because they do not focus on describing an event and only mention a relevant event briefly in the document. For example, the MUC-4 corpus includes interviews, speeches, and terrorist propaganda that contain information about terrorist events. The categorizing of documents that mention events into event narratives and fleeting references is a general observation across different types of events. Figure 3.3 and 3.4 show examples of an event narrative and a fleeting reference accordingly for civil unrest events. Specifically, the story as shown in Figure 3.4 is an event narrative about an attack, but contains a fleeting reference to a civil unrest event. Instead of manifesting each piece of event information as in Figure 3.1 and 3.2, the underlined sentences refer to the parts of documents that contain event information.

## 3.2 TIER: Zooming in on Event Information

The main idea behind my approach is to analyze documents at multiple levels of granularity in order to identify role fillers that occur in different types of contexts. My

BOGOTA, 24 AUG 89 (EFE) -- [TEXT] THE PRESENCE OF BRITISH AND ISRAELI MERCENARIES, PAID BY COLOMBIAN EXTREME RIGHT-WING GROUPS AND DRUG TRAFFICKERS, WAS CONFIRMED TODAY BY THE ADMINISTRATIVE DEPARTMENT OF SECURITY (DAS), COLOMBIA'S SECRET POLICE.

THE DAS SENT A CONFIDENTIAL REPORT TO COLOMBIAN PRESIDENT VIRGILIO BARCO WHICH WAS PUBLISHED TODAY BY THE LIBERAL NEWSPAPER EL TIEMPO. THE REPORT CONFIRMS THAT THERE IS A COLLABERATION BETWEEN DRUG TRAFFICKERS AND EXTREME RIGHT-WING GROUPS, AND THE PRESENCE OF FOREIGN MERCENARIES IN COLOMBIA -- HIRED BY THE EXTREME RIGHT-WING DRUG-TRAFFICKING ORGANIZATIONS.

THE OFFICIAL ORGANIZATION ASSERTS THAT THE RECENTLY CREATED EXTREME RIGHTIST MOVEMENT OF NATIONAL RESTORATION (MORENA) "MUST BE IDENTIFIED AS THE CULMINATION OF A SYMBIOSIS BETWEEN DRUG TRAFFICKERS, SELF-DEFENSE GROUPS, GROUPS OF HIRED GUNMEN TRAINED BY FOREIGN MERCENARIES, AND AGRARIAN BUSINESSMEN WHO ARE TIRED OF THE GUERRILLA GROUPS' CONSTANT HARASSMENT."

THE REPORT INDICATES THAT BRITISH AND ISRAELI MERCENARIES *IDENTIFIED AS* JOHN OWEN, DAVE TOMKINS, R. PAXTON, P. GLASGOW, A. DEWER, AND P. ATHERTON ENTERED COLOMBIA THROUGH ELDORADO INTERNATIONAL AIRPORT IN JULY, BUT THE INFORMATION DOES NOT GIVE THEIR INDIVIDUAL NATIONALITIES.

THE FIRST PERSONS TO NOTIFY THE PROSECUTOR GENERAL OF THE NATION, THE HIGHEST AUTHORITY, ABOUT THE PRESENCE OF FOREIGN MERCENARIES AMONG THE DRUG-TRAFFICKERS FAR RIGHT-WING ARMED GROUPS WERE DESERTERS FROM THE MIDDLE MAGDALENA SELF-DEFENSE GROUPS -- *IDENTIFIED AS* JESUS ALBERTO MOLINA URREA, RICAURTE DUQUE ARBOLEDA, AND VICTOR ARBOLEDA DUQUE.

ACCORDING TO THE DESERTERS' DECLARATIONS, THE ISRAELI MERCENARIES CANNOT SPEAK SPANISH AND THEY FREQUENTLY MET WITH GONZALO DE JESUS PEREZ, HENRY PEREZ, AND MARCELO PEREZ, WHO, ACCORDING TO THE DAS REPORT REVEALED TODAY, ARE IMPORTANT MEMBERS OF THE ASSOCIATION OF MIDDLE MAGDALENA CATTLEMEN (ACDEGAM), THE SELF-DEFENSE GROUPS, AND MORENA; AND HAVE BEEN CHARGED WITH *THE MASS MURDERS OF* PEASANTS IN URABA, LOCATED IN THE COUNTRY'S NORTHWESTERN AREA.

**Figure 3.2**: Another Terrorism Event Story (ID: TST1-MUC3-0026, excerpted)

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 that would have been otherwise missed. This multilayered approach creates an event extraction system that can discover role fillers in a variety of different contexts, while maintaining good precision.

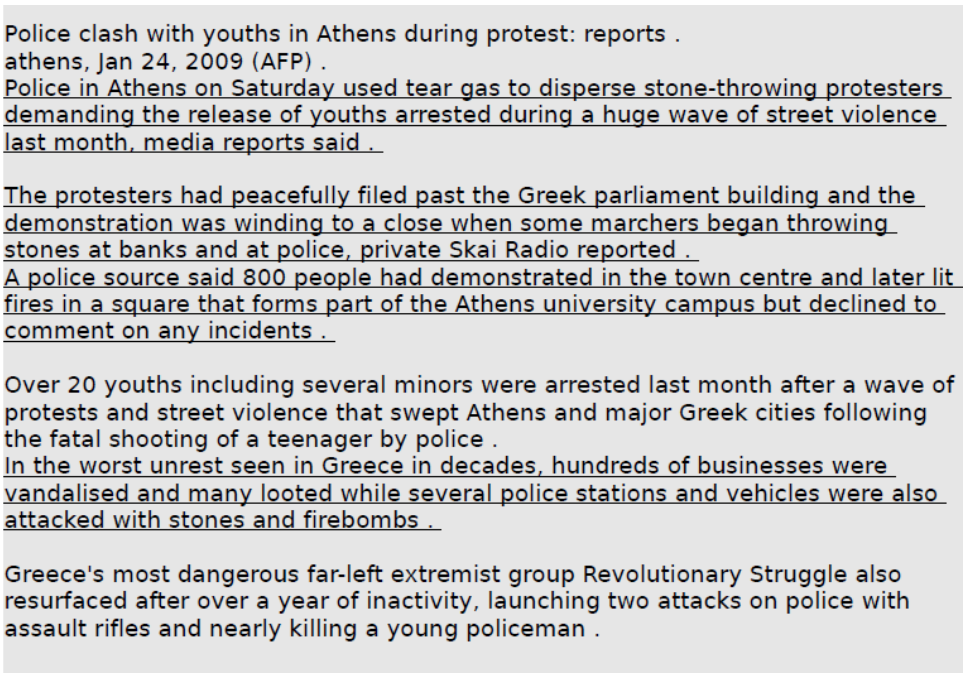


**Figure 3.3**: An Event Narrative Story about Civil Unrest

To accurately detect secondary event contexts, I adopt a two-pronged strategy for event extraction that handles *event narrative* documents differently from other documents. I 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. To ensure that relevant information is extracted from all documents, I also apply a conservative extraction process to every document to extract facts from explicit event sentences.

My complete event extraction model, called TIER (as shown in Figure 3.5), 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, I train a text genre classifier to identify event narrative documents. At the middle level, I 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, I 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.

Israel launches manhunt after deadly attack on gay club .
tel aviv, Aug 2, 2009 (AFP) .
Israeli police were hunting on Sunday for a masked man who opened fire at a gay youth club in Tel Aviv, killing two people in an attack that struck fear among the liberal city's homosexual community .

The black-clad gunman used a pistol to target the group of young gays and lesbians at the entrance to the community centre in the heart of Israel's commercial capital late on Saturday and then fled, police and witnesses said .
A teenage girl and a man in his 20s were killed on the spot and 15 people were wounded, three seriously, police said, adding that a manhunt has been launched for the assailant .

It was the worst attack against country's gay and lesbian community .
Thousands of people gathered in the centre of the Mediterranean seaside city overnight to protest against the attack on the Bar Noar ("Youth Bar"), some waving rainbow banners and lighting candles for the victims .

The victims were identified as Liz Tarbishi, 17 and Nir Katz, 26 .

Israeli Prime Minister Benjamin Netanyahu condemned the "shocking murder" and called on the police to do everything to bring the gunman to justice .
Gays and lesbians enjoy freedom in Israel, serving openly in the military, and Tel Aviv holds an annual gay pride parade .
Israel repealed a ban on consensual same-sex sexual acts in 1988 and certain rights of gay or lesbian couples have since been recognised by the courts .

**Figure 3.4**: A Story with a Fleeting Reference about Civil Unrest

## 3.3 Stratified Extraction: Two Extraction Paths

An important aspect of my 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 main topic of the article. Consequently, in the first extraction path as illustrated in Figure 3.6, 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.

The second path (as shown in Figure 3.7) 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 the following subsections, I explain how I create the two types of sentence classifiers and the role filler extractors. I will return to the issue of document genre and the event narrative classifier in Section 3.4.3.

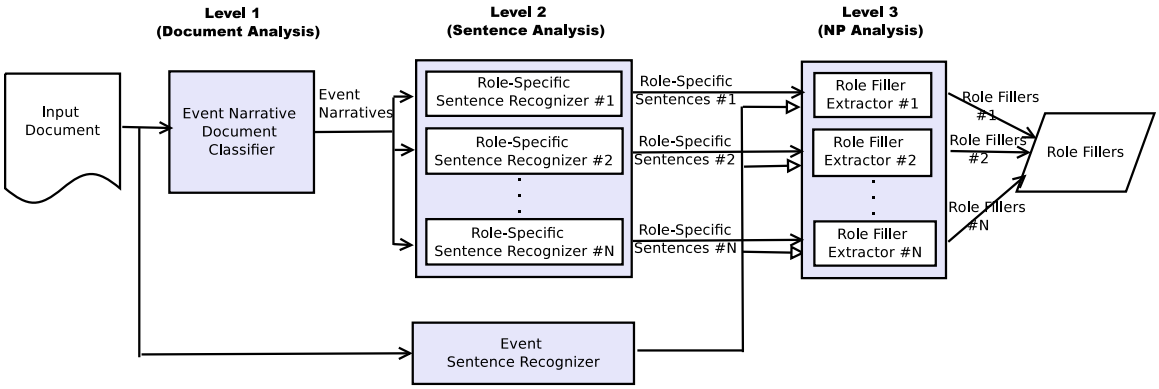


**Figure 3.5**: TIER: A Multilayered Architecture for Event Extraction

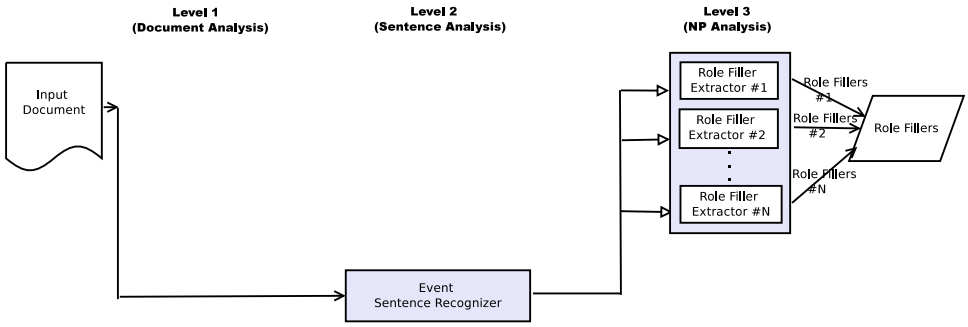


**Figure 3.6**: The First Extraction Path to Process Primary Event Contexts

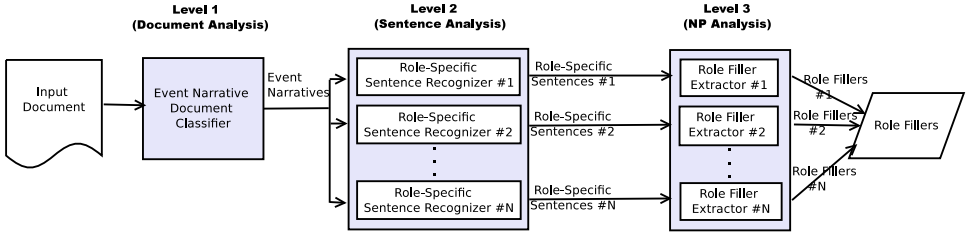


**Figure 3.7**: The Second Extraction Path to Process Secondary Event Contexts

## 3.4 Implementation Details

### 3.4.1 Sentence Classification

I 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, I use sentences as my 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 may precede or follow the event.

For both types of classifiers, I use exactly the same feature set, but I train them in different ways. Generally in event extraction annotations, each document that describes a relevant event has answer key templates with the role fillers (*answer key strings*) for relevant events. To train the event sentence recognizer, I consider a sentence to be a positive training instance if it contains one or more answer key strings from any of the event roles. All remaining sentences that do not contain any answer key strings are used as negative instances.

There is no guarantee that a classifier trained in this way will identify event sentences, but my 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 [102], and they demonstrated that this approach worked reasonably well when compared to training with event sentences labelled by human judges.

The main contribution of my work is introducing additional *role-specific sentence classifiers* to seek out role fillers that appear in less obvious secondary contexts. I 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. I 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. In this way, I force each classifier to focus on the contexts specific to its particular event role. I 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, I randomly choose from the negative instances to produce the same ratio of negative to positive instances as the event sentence classifier.

Both types of classifiers use an SVM model created with SVMlin [55], 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[1], 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.

[1]I used the Sundance parser [99] to identify noun phrases and assign semantic class labels.

### 3.4.2 Role Filler Extractors

My 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. I randomly choose among all noun phrases that are not in the answer keys to create a 10:1 [2] ratio of negative to positive instances.

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. I will refer to the noun phrase being labelled as the *targeted NP*. The role filler extractors use three types of features:

1 Lexical features

I 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.

2 Lexico-syntactic patterns

I use the AutoSlog pattern generator [94] 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).

3 Semantic features

I use the Stanford NER tagger [29] to determine if the targeted NP is a named entity, and I use the Sundance parser [99] to assign semantic class labels to each NP's head noun.

### 3.4.3 Event Narrative Document Classification

One of my goals was to explore the use of *document genre* to permit more aggressive strategies for extracting role fillers. In this section, I first present an analysis of one of my experimental data sets, the MUC-4 data set, a standard benchmark collection of terrorism event stories that are used for evaluating event extraction systems, which reveals

[2] This ratio was determined empirically by optimising performance on the tuning data; it may need to be adjusted for unseen domains.

the distribution of event narratives in the corpus. Next, I will explain how I train a classifier to automatically identify event narrative stories.

#### 3.4.3.1 Manual Analysis

I define an *event narrative* as an article whose main focus is on reporting the details of an event. For the purposes of this research, I am 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. I 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 I might create a system to automatically distinguish event narrative documents from fleeting reference documents, I manually labelled the 116 relevant documents in the tuning set. This was an informal study solely to help us understand the nature of these texts.

The first row of Table 3.1 shows the distribution of event narratives and fleeting references based on my "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.

#### 3.4.3.2 Heuristics for Event Narrative Identification

My 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 my manual analysis of the tuning set, I developed several heuristics to help separate them.

**Table 3.1**: Manual Analysis of Document Types

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

I observed two types of clues: the location of the relevant information, and the density of relevant information. First, I 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, my first heuristic requires that an event narrative mention a role filler within the first several sentences.

Second, event narratives generally have a higher density of relevant information. I use several criteria to estimate information density because a single criterion was inadequate to cover different scenarios. 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. I 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 3.8 shows histograms for different values of this ratio in the event narrative (a) vs. the fleeting reference documents (b) in the MUC-4 data set. The histograms clearly show that documents with a high ($> 50\%$) ratio are almost always event narratives.

A second density heuristic requires that the ratio of different *types* of roles 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. Experiments showed that heuristic #1 covers most of the event narratives.

I use these heuristics to label a document as an event narrative if: (1) it has a high density

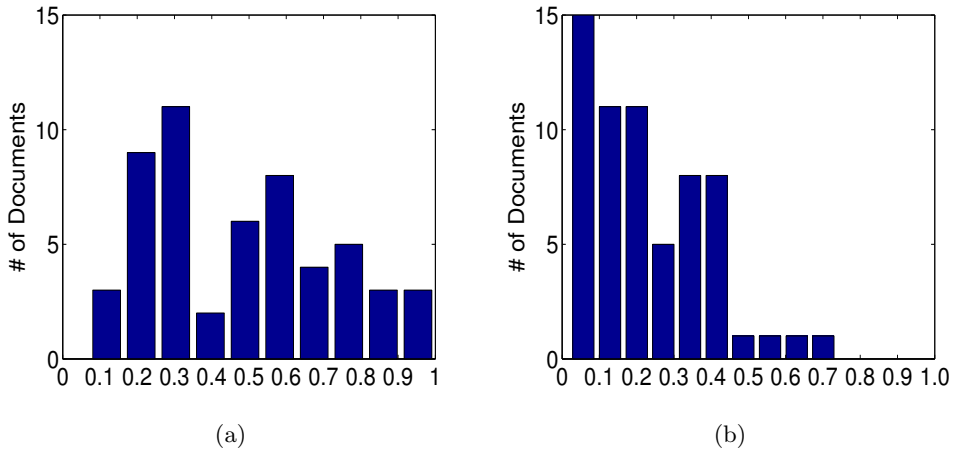


**Figure 3.8**: Histograms of Relevant Sentence Densities in Event Narratives (a) and Fleeting References (b)

of relevant information, AND (2) it mentions a role filler within the first 7 sentences.

The second row of Table 3.1 shows the performance of these heuristics on the tuning set in the MUC-4 data 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.

#### 3.4.3.3 Event Narrative Classifier

The heuristics above use the answer keys to help determine whether a story belongs to the event narrative genre, but my goal is to create a classifier that can identify event narrative documents without the benefit of answer keys. So I 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 not. In the MUC-4 data set, of the 700 relevant documents, 292 were labeled as event narratives. I 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., On the tuning data, my heuristics failed to identify 25% of the event narratives.). I then trained an SVM classifier using bag-of-words (unigram) features.

Table 3.2 shows the performance of the event narrative classifier on the manually labeled tuning set in the MUC-4 data set. The classifier identified 69% of the event narratives with 63% precision. Overall accuracy was 81%.

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

**Table 3.2**: Event Narrative Classifier Results

| Recall | Precision | Accuracy |
|---|---|---|
| .69 | .63 | .81 |

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 something relevant.

#### 3.4.3.4 Domain-relevant Document Classifier

For comparison's sake, I also created a document classifier to identify *domain-relevant* documents. That is, I trained a classifier to determine whether a document is relevant to the domain of terrorism, irrespective of the style of the document. I 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. I use this classifier for several experiments described in the next section.

## 3.5 Evaluation

### 3.5.1 Data Sets

To verify the general applicability of my multilayered event extraction architecture to extract events of different types, I will evaluate the implemented system on two data sets of distinct event domains.

The first one is the MUC-4 data set [76], a standard benchmark collection for evaluating event extraction systems. The corpus consists of 1700 documents about Latin American terrorist events including kidnapping, arson, bombing, and other attack events. Each document comes with associated answer key templates, a template per event. Roughly half of the documents are relevant (i.e., they mention at least 1 terrorist event) and the rest are irrelevant.

The second domain is for civil unrest events. Civil unrest (CU) is a broad term that is typically used to describe a form of public disturbance caused by a group of people for a purpose. Types of civil unrest can include strikes, rallies, sit-ins, and other forms of obstructions, riots, sabotage, and other forms of public disturbance motivated by a cause. I created a new civil unrest data set for this research. I defined initial human annotation guidelines and modified them in several iterations to address the confusions and issues that the annotators came across when they applied the guidelines. The annotated documents were selected from the English Gigaword corpus [83], by randomly sampling from the documents that contain one of six predefined civil unrest event keywords: "protest", "strike", "march", "rally", "riot", and "occupy", or their morphological variations. Note that I used "marched" and "marching" as keywords but did not use "march" because it often refers to a month.

#### 3.5.1.1 Creating Civil Unrest Event Annotations

The annotations were obtained through two stages. First, at the document level, two human annotators identified the documents that mention a civil unrest event, following the guidelines as specified in Appendix A. In this stage, the two annotators first annotated 100 documents in common and they achieved a relatively high $\kappa$ [25] score of .82. Cohen's kappa coefficient is a statistical measure of interrater agreement. It is generally thought to be a more robust measure than simple percent agreement calculation because $\kappa$ takes into account the agreement occurring by chance. Then, each annotator annotated 150 more documents. Therefore, in total, 400 documents were annotated in this stage. Out of 400 documents, 129 documents were labeled as event relevant (i.e., mentioning a civil unrest event). Therefore, around two thirds of the documents that contain event keywords did not mention any civil unrest event.

In the second stage, the documents that were labeled as event-relevant in the previous stage were additionally labeled with respect to identified event roles for civil unrest events. Before event role filler annotations, I removed summary articles first and then had annotators label the rest documents. Summary articles are essentially a list of news summaries and do not elaborate on any particular story. Specifically, I removed 28 summary articles; among these, 14 was labeled as event-relevant. Therefore, 372 documents will be used for evaluating my event extraction systems. The event role filler annotation guidelines are specified in Appendix B. Overall, six event roles were considered for annotations. They are *agents* of CU events, *sites* and *locations* where events occur, *victims* and *(affected) facilities*, and *instruments* that are used during CU events. Figure 3.9 shows the event role filler annotations of the document as we have seen in Figure 3.3, where the underlined sentences are event relevant. Over the six event roles, the annotators achieved an overall $\kappa$ score of .83. Then, the two annotators adjudicated their decisions to create the final civil unrest event annotations.

### 3.5.2 Evaluation Methods

For the MUC-4 data set, I evaluate the implemented system on the five "string-fill" event roles: *perpetrator individuals, perpetrator organizations, physical targets, victims,* and *weapons*. Table 3.3 shows the distribution of gold role fillers in the MUC-4 test set. For the civil unrest data set, I evaluate the system on four event roles: *agents, sites, locations,* and *instruments*. Two other event roles, *victims* and *(affected) facilities*, were annotated too, but I decided not to include them in the evaluation because they are only sparsely seen in civil unrest descriptions. Table 3.4 shows the distribution of gold role fillers in the Civil

**EVENT TYPE(S):** MARCH
**AGENT:**
youths
stone-throwing protesters
The protesters
some marchers
800 people
**SITE:**
the town centre
the Greek parliament building
a square
**LOCATION:**
Athens
Greece
**VICTIM:**
**(Affected) FACILITY:**
several police stations
vehicles
hundreds of businesses
**INSTRUMENT:**
tear gas
stones
firebombs

**Figure 3.9**: An Example: Civil Unrest Event Annotations

**Table 3.3**: # of Role Fillers in the MUC-4 Test Set

| PerpInd | PerpOrg | Target | Victim | Weapon |
|---|---|---|---|---|
| 129 | 74 | 126 | 201 | 58 |

**Table 3.4**: # of Role Fillers in the CU Test Set

| Agents | Sites | Locations | Instruments |
|---|---|---|---|
| 408 | 120 | 147 | 67 |

Unrest data set.

The complete IE task involves template generation, one template per event, which is complex because many documents have multiple templates (i.e., they discuss multiple events). My work focuses on extracting individual facts and not on template generation per se (e.g., I do not perform coreference resolution or event tracking). Consequently, my 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"). Pronouns were discarded since I do not perform coreference resolution. Duplicate extractions with the same head noun were counted as one hit or one miss.

### 3.5.3 Metrics

My results are reported as *Precision/Recall/F(1)-score* for each event role separately. *Precision* measures the accuracy of event extraction systems and it is defined as the ratio of the correct role filler extractions over the total number of extractions generated by a system. *Recall* measures the coverage of event extraction systems. It is defined as the ratio of the correct role filler extractions over the total number of gold extractions that are annotated by humans. *F(1)-score* is the harmonic mean of *Precision* and *Recall*. The following defines *F(1)-score*:

$F(1)score = \frac{2*Precision*Recall}{Precision+Recall}$

I also show an overall average for all event roles combined. For the previous systems to which I compare my systems, I generated the Average scores myself by macro-averaging over the scores reported for the individual event roles.

## 3.6 Evaluating TIER on the MUC-4 Data Set

In this section, I will show experimental results on the MUC-4 data set. MUC-4 data is a standard benchmark collection and it has been used to evaluate several previous event extraction systems. Therefore, for this data set, I can compare the performance of my system with three other event extraction systems that have reported evaluation results on this data set. To be consistent with previously reported results, out of the total 1700 documents, I use the 1300 DEV documents for training, 200 documents (TST1+TST2) as the tuning set, and 200 documents (TST3+TST4) as the test set.

In addition to reporting the results of my multilayered event extraction system, I will also evaluate its variations by replacing or taking off certain components of the full system. Finally, based on the performance of my system, I will present my analysis on the tuning documents of the MUC-4 data set, shedding light on the strengths and limitations of TIER.

### 3.6.1 Baselines

As baselines, I compare the performance of my IE system with three other event extraction systems. The first baseline is AutoSlog-TS [95], 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 [85] *semantic affinity* model (PIPER) and the [102] GLACIER system. The *semantic affinity* approach 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. Please also refer to Patwardhan's Ph.D. dissertation [84] for more details.

The first 3 rows in Table 3.5 show the results for each of these systems on the MUC-4 test set. They all used the same evaluation criteria as my results.

### 3.6.2 Experimental Results

The middle portion of Table 3.5 shows the results of a variety of event extraction models that I created using different components of my system. The **AllSent** row shows the performance of my 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 my Role Filler Extractors applied only to the *event sentences* identified by my event sentence classifier. This boosts precision across all event roles, but with a sharp reduction in recall. There is a roughly 20 point swing from recall to precision. These results are similar to GLACIER's results on most event roles, which is not surprising because GLACIER also incorporates event sentence identification.

The **RoleSent** row shows the results of my Role Filler Extractors applied only to the *role-specific sentences* identified by my classifiers. There is a 12-13 point swing from recall to precision compared to the **AllSent** row. 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. I see slightly higher recall, which confirms that one set of extractions is not a strict subset of the other. However, precision is still relatively low.

The next set of experiments incorporates document classification as the third layer of text analysis. Here, I wanted to determine how event narrative document classification

**Table 3.5**: Experimental Results on the MUC-4 Data Set, Precision/Recall/F-score

| 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 | | | | | | |
| **Dom/(ESent+RSent)** | 45/54/49 | 42/51/46 | 51/68/58 | 54/56/55 | 46/63/53 | 48/58/52 |
| **ESent+Dom/RSent** | 43/59/50 | 45/61/**52** | 51/77/**61** | 52/61/56 | 44/66/53 | 47/65/54 |
| **ESent+ENarr/RSent** | 48/57/52 | 46/53/50 | 51/73/60 | 56/60/**58** | 53/64/**58** | 51/62/**56** |

performed compared to topic-based document classification, as used in the multilayered event extraction implementations. Therefore, I trained two different document classifiers. The Event Narrative Document Classifier (**ENarr**) was trained to identify event narratives, which are documents that are dedicated to report details of events. In contrast, the Domain-relevant Document Classifier (**Dom**), as described in Section 3.4.3.4, was trained to determine whether a document is relevant to the domain and describes any relevant event, irrespective of the style of the document. The **Dom/(ESent+RSent)** row shows the results of applying both types of sentence classifiers only to documents identified as domain-relevant by the Domain-relevant Document Classifier. Extracting information only from domain-relevant documents improves precision by +6, but also sacrifices 8 points of recall.

The **EventSent** row revealed that information found in event sentences has the highest precision, even without relying on document classification. I concluded that evidence of an event sentence is probably sufficient to warrant role filler extraction irrespective of the style of the document. As I discussed in Section 3.4.3, 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, I created a system, **ESent+Dom/RSent**, 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, I evaluated the idea of using document *genre* as a filter instead of domain relevance. The last row, **ESent+ENarr/RSent**, shows the results of my 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. Therefore, compared to the Domain-relevant Document Classifier, event narrative genre recognition is more effective to seek out secondary event contexts, when plugged in the multilayered event extraction architecture.

Overall, TIER's multilayered 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 my 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 informa-

tion only from event sentences in all other documents.

### 3.6.3 Analysis

I 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 during 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.

## 3.7 Evaluating TIER on the Civil Unrest Data Set

Compared to the MUC-4 corpus (1700 documents), the Civil Unrest data set (372 documents) is much smaller. Therefore, for this data set, I will report the 10-fold cross-validation results. Similar to the evaluation for the MUC-4 corpus, for the civil unrest data set, I will also evaluate both my multilayered event extraction system and its variations by replacing or taking off certain components of the full system. Then, concerned with the limited size of this data set, I will show the learning curve of my full multilayered event extraction system by running the system on a quarter of the data and increasing the data by another quarter per run.

### 3.7.1 Experimental Results

The first section of Table 3.6 shows the performance before incorporating document classification. The first row **AllSent** shows the results of applying role filler extractors only. We can see that without the benefits of high level contextual analysis components, the local role filler extractors are not so precise and the overall extraction precision is only .24. The second row **EventSent** shows that by only applying the role filler extractors within event sentences as identified by the event sentence classifier, the extraction accuracy was greatly improved to .49, but the recall was reduced by half from .48 to .24. The third row **RoleSent** shows that by extracting information from the role specific contexts as identified by the role-specific sentence classifiers, the precision is 33% while more extractions were found compared to using the event sentence classifier filter. The fourth row **EventSent+RoleSent** shows that if we extract role fillers from both event sentences and role-specific sentences, we achieve further gain in recall which implies that the event sentences identified by the event sentence

**Table 3.6**: Experimental Results on the Civil Unrest Data Set, Precision/Recall/F-score

| Method | Agent | Site | Location | Instrument | Average |
|---|---|---|---|---|---|
| Results without document classification | | | | | |
| **AllSent** | 37/51/43 | 23/38/28 | 13/49/21 | 44/70/54 | 29/52/38 |
| **EventSent** | 62/25/35 | 50/19/28 | 39/15/22 | 75/57/64 | 56/29/38 |
| **RoleSent** | 45/43/44 | 32/24/27 | 20/31/24 | 61/54/57 | 39/38/39 |
| **EventSent+RoleSent** | 45/43/44 | 35/32/33 | 20/32/25 | 60/64/62 | 40/43/41 |
| Results with document classification | | | | | |
| **Dom/(ESent+RSent)** | 46/38/42 | 38/28/33 | 28/29/28 | 65/61/63 | 44/39/42 |
| **ESent+Dom/RSent** | 47/41/44 | 39/31/35 | 27/30/29 | 66/64/65 | 45/41/43 |
| **ESent+ENarr/RSent** | 50/39/44 | 41/28/34 | 30/29/29 | 67/64/66 | 47/40/43 |

classifier are not strictly a subset of the sentences labeled by the role-specific sentences. However, the precision is still low at 33%.

The second section of the table shows the performance of the event extraction systems after incorporating the document classification components. The first row here shows that the domain document classifier, as described in Section 3.4.3.4, helps to improve the extraction precision on top of the sentential classifiers with a small reduction in recall. The second row **ESent+Dom/RSent** shows the performance when the domain document classifier was only applied on top of the role-specific classifiers. Compared to the results as in the first row, the precision was the same which means that the event sentences can be safely applied to identify event sentences from all the documents. The recall was slightly increased because the event sentence classifier found event sentences from the documents that were not labeled as domain-relevant. The last row **ESent+ENarr/RSent** shows the superior precision achieved after replacing the domain document classifier with the event narrative document classifier, with one point of recall loss.

Overall, similar to what we have observed from the evaluation results using the MUC-4 corpus, the *role-specific sentence classifiers* help to recover event role filler information that is missed by the event sentence classifier. In addition, limiting the application of *role-specific sentence classifier* within event narratives as identified by the *event narrative document classifier* improves precision.

### 3.7.2 Learning Curve

Compared to the evaluation results on the MUC-4 corpus, the performance of *TIER* is relatively low on the Civil Unrest data set. This is prabably due to the limited size of the data set. To show how the extraction performance was affected by the size of data, I drew the learning curve by running the system on different proportions of the data set. Specifically, I start with running the system on a quarter of the data. Then, I incrementally

add in more data, one quarter a time, and run *TIER* on the gradually enlarged data set. From the learning curve as shown in Figure 3.10, we can see that the extraction performance of *TIER* was clearly improved with more and more data fed in.

## 3.8 Conclusions

In this chapter, I discussed the design and details of my multilayered event extraction architecture, *TIER*, which incorporates both document genre and role-specific context recognition into 3 layers of analysis to seek out event formation in a variety of different contexts. Experimental results on two event domains show that *TIER* can recover more event information compared to previous event extraction systems, while maintaining a good extraction precision.

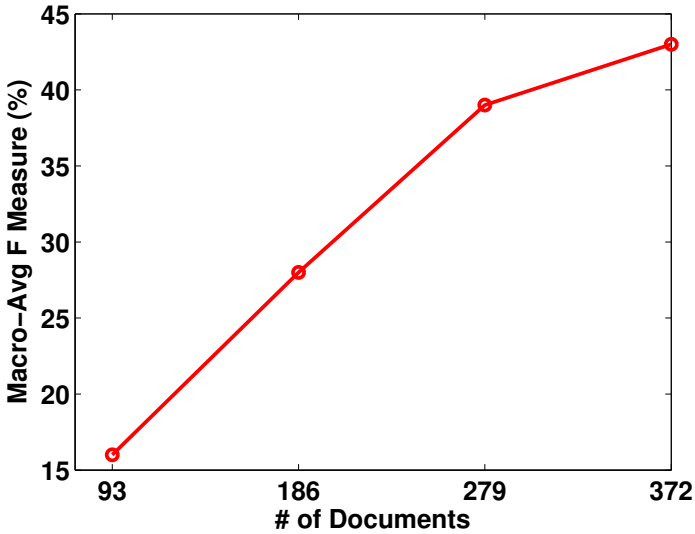


**Figure 3.10**: Learning Curve for TIER

# CHAPTER 4

# *LINKER*: A DISCOURSE-GUIDED ARCHITECTURE

As described in the previous section, TIER focuses on improving the recall of event extraction performance by seeking out event information from secondary contexts. While TIER is conservative when extracting information from secondary contexts by limiting the extraction from secondary contexts within event narrative documents only, TIER essentially assumes that the primary contexts that mention event keywords or event phrases are reliable and will always be examined further for extraction purposes. However, depending on the larger context, the seemingly relevant local context may not be referring to a relevant event due to ambiguity and metaphor. For example, "Obama was attacked" may lead to Obama being extracted as the victim of a physical attack, even if the preceding sentences describe a presidential debate and the verb "attacked" is being used metaphorically. Therefore, both primary contexts and secondary contexts needed to be validated and strengthened by looking beyond the current sentence and incorporating contextual influence from a wider discourse, including the preceding and following sentences of the current sentence.

By design, TIER uses two types of sentence classifiers, *event sentence classifier* and *role-specific sentence classifier*, to identify event information occuring in a variety of different event contexts. In addition, observing that event descriptions in event narratives and fleeting references are different in nature, TIER includes a document classifier to identify event narratives too. Together with the set of local role filler extractors, the four components are responsible to analyze texts in multiple granularities. Note that all the components can be trained independently; therefore, one unique feature of TIER is that it is well modulized and each component is easily trained. Logically, TIER distributes the text processing burden to four components and arrange them in a novel way to extract event information effectively.

However, due to the modularity, *TIER* is incapable of capturing content flows in the discourse level. To address the limitations of *TIER*, I will present my bottom-up event extraction architecture, called *LINKER*, that can explicitly model textual cohesion properties

across sentences. *LINKER* includes a single sequentially structured sentence classifier that identifies event-related story contexts. The sentence classifier uses lexical associations and discourse relations across sentences, as well as domain-specific distributions of candidate role fillers within and across sentences to identify all the event contexts.

In the following sections, I will first depict the bottom-up design of *LINKER*, then I will describe in detail the different types of features that are used in the structured sentence classifier. Finally, I will present the evaluation results of *LINKER* on the same two event extraction data sets as used in Chapter 3.

## 4.1 LINKER: A Bottom-up Extraction Model

To model contextual influences across sentences, I propose a bottom-up approach for event extraction, called *LINKER*, that aggressively identifies *candidate role fillers* based on local (intrasentential) context, and then uses distributional properties of the candidate role fillers as well as other discourse features to model textual cohesion across sentences. This event extraction architecture has two components: (1) a set of local role filler extractors, and (2) a sequential sentence classifier that identifies event-related story contexts. The novel component is the sentence classifier, which uses a structured learning algorithm, conditional random fields (CRFs), and features that capture lexical word associations and discourse relations across sentences, as well as distributional properties of the candidate role fillers within and across sentences. The sentence classifier sequentially reads a story and determines which sentences contain event information based on both the local and preceding contexts. The two modules are combined by extracting only the candidate role fillers that occur in sentences that represent event contexts, as determined by the sentence classifier.

My event extraction model (see Figure 4.1) involves two processes that each focus on a different aspect of the problem. The left side of Figure 4.1 shows the two components and illustrates how they interact. The top component on the left is a set of traditional role filler detectors, one for each event role. This component identifies candidate role fillers based on the immediate context surrounding a noun phrase. These role fillers tend to be overly aggressive on their own, producing many correct extractions but also many false hits.

The bottom component on the left side of Figure 4.1 is a structured sentence classifier that identifies event-related story contexts. This classifier determines whether a sentence is discussing a domain-relevant event based on two types of information. The structured learning algorithm explicitly considers whether the previous sentence is an event context when classifying the next sentence, which captures discourse continuity across sentences. I also provide the learner with features representing other textual cohesion properties, in-

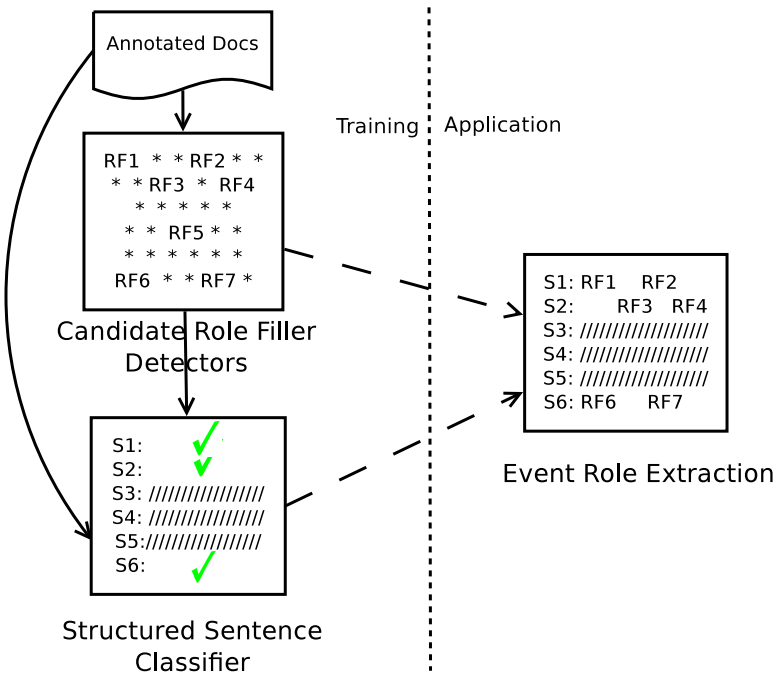


**Figure 4.1**: A Bottom-up Architecture for Event Extraction

cluding lexical associations and discourse relations between adjacent sentences. In addition, the bottom-up design of the architecture provides information about candidate role fillers found by the local detectors. This domain-specific information is incorporated into features that represent the number, types, and distribution of the candidate role fillers both within and across sentences.

The two components provide different sources of evidence that are both considered when making final extraction decisions. The right side of Figure 4.1 illustrates how the two components are used. The event extraction system only produces a role filler if the noun phrase was hypothesized to be a candidate role filler based on local context *and* it appears in an event-related story context, as determined by the sequential sentence classifier. In the following sections, I describe each of these components in more detail.

### 4.1.1 Candidate Role Filler Detectors

The mission of the candidate role filler detectors is to analyze each noun phrase and identify candidate role fillers using their local contextual clues. As shown in Figure 4.2, the candidate role filler detectors will analyze each noun phrase (represented as a *) in a document independently and classifier it with respect to an event role. Our candidate role filler detectors are identical to the local role filler extractors used by TIER [44], which allows for direct comparisons between TIER and our new model. They are also very similar to the

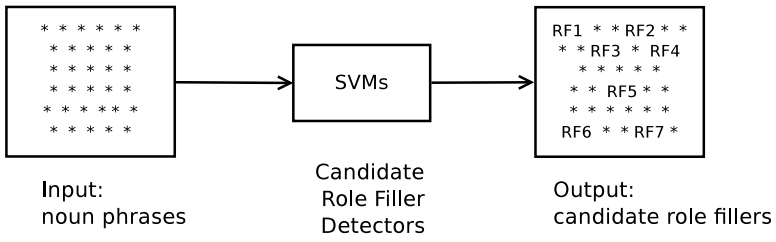


**Figure 4.2**: Candidate Role Filler Extraction Process.

plausible role filler detectors used by GLACIER [102] (the other system we compare against in Section 4.3), except for small differences in the lexical features and the positive/negative training ratios.

### 4.1.2 Structured Sentence Classification

The sequential sentence classifier is responsible for determining which sentences are related to domain-relevant events. I utilize conditional random fields (CRFs) [59] to carry out this sequential labeling task. A sequential CRF is a structured discriminative learning model that produces a sequence of labels using features derived from the input sequence. This component will sequentially read the sentences in a story and determine whether each sentence is discussing a relevant event based on direct evidence from both the current sentence and the previous sentence. All other sentences only affect the results indirectly through label transitions.

As shown in Figure 4.3, given a whole document as input, the structured sentence classifier classifies each sentence with respect to a particular type of event while consulting the evidence coming from surrounding sentences. As a result, the structured classifier will produce a sequence of labels in a single pass, one per sentence, to indicate if the sentence describes an event or not. Each label in the output sequence is binary. "1" indicates that its corresponding sentence describes an event while "0" indicates that the sentence does not describe any relevant event.

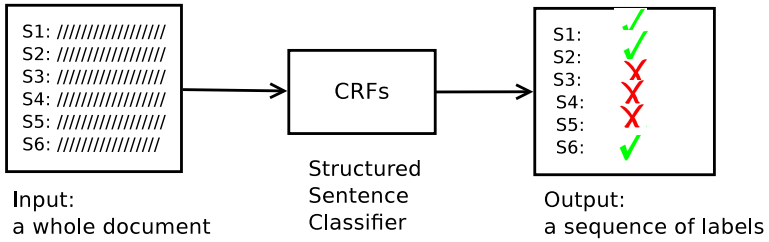


**Figure 4.3**: Structured Sentence Classifier: Finding Event-related Contexts.

I used the CRF++ toolkit (http://crfpp.sourceforge.net/#tips [crfpp.sourceforge.net]) to create our structured sentence classifier. CRF++ performs sequential labeling tasks and requires each unit in the input to have a fixed number of raw features. Since the length of sentences can vary, affecting the number of n-grams and other features accordingly, I expand the feature vector for each sentence with pseudo-tokens[1] as needed to ensure that every sentence has the same number of features. The toolkit was modified not to generate real features from the pseudo-tokens.

## 4.2 Features for the Structured Sentence Classifier

I provide the classifier with rich types of linguistically motivated features to represent individual sentences and textual cohesion properties linking adjacent sentences: basic features, lexical bridges, discourse bridges, and role filler distributions. The following subsections describes each of these feature sets in detail.

### 4.2.1 Basic Features

As the basic representation of a sentence, I use unigram and bigram features. I create features for every unigram and bigram, without stemming or stopword lists. In addition, I found it beneficial to create five additional features representing the first five bigrams in the sentence. I define features for positions 1 through 5 of a sentence to represent the bigrams that begin in each of these positions. I hypothesize that these positional bigram features help to recognize expressions representing discourse cue phrases at the beginning of a sentence, as well as the main subject of a sentence.

### 4.2.2 Lexical Bridge Features

An important aspect of textual cohesion is lexical word associations across sentences. This idea has been explored in [109] to model the intuition that the use of certain words in a discourse unit (e.g., sentence) tends to trigger the use of other words in subsequent discourse units. In the context of event extraction, a pair of related event keywords may occur in consecutive sentences. For example, it is common to see "bombed" in one sentence and "killed" in the next sentence because bombing event descriptions are often followed by casualty reports. Similarly, we may see "attacked" and "arrested" in adjacent sentences because a mention of an attack is often followed by news of the arrest of suspected perpetrators.

---

[1] I define a special token for this purpose.

To capture lexical associations between sentences, I create *lexical bridge* features that pair each verb in the current sentence ($Verb_i$) with each verb in the preceding sentence sentence ($Verb_{i-1}$):

$$< Verb_{i-1}, Verb_i >$$

To obtain better generalization, I stem the verbs before creating the bridge features using the Porter stemmer [89]. For example, a sentence that mentions a bombing followed by a sentence containing "killed" would generate the following lexical bridge feature:

$$< bomb, kill >$$

Event keywords could also appear as nouns, such as "assassination" and "death". Therefore, I also create lexical bridge features by pairing nouns from the current sentence and the preceding sentence:

$$< Noun_{i-1}, Noun_i >$$

For example, if we see the word "explosion" in the preceding sentence and the nouns "people" and "offices" in the current sentence, then two features will be created as follows:

$$< explosion, people >$$

$$< explosion, offices >$$

I also tried including associations between nouns and verbs in adjacent sentences (i.e., $< Verb_{i-1}, Noun_i >$ and $< Noun_{i-1}, Verb_i >$), but they did not improve performance. To focus on event recognition, the lexical bridges are only created between sentences that each contain at least one candidate role filler.

### 4.2.3 Discourse Bridge Features

I also represent two types of discourse relations between consecutive sentences: discourse relations produced by a Penn Discourse Treebank (PDTB) trained discourse parser, and syntactic discourse focus relations. I hypothesized that these features could provide additional evidence for event label transitions between sentences by recognizing explicit discourse connectives or a shared discourse focus.

PDTB-style discourse relations [90] are organized hierarchically in three levels based on different granularities. I use the discourse relation output produced by a PDTB-style discourse parser [65]. Given a text, the discourse parser generates both explicit (triggered by cue phrases such as "if" or "because") and implicit level-2 PDTB discourse relations,

such as cause, condition, instantiation, and contrast. A discourse relation may exist within a sentence or between two adjacent sentences in the same paragraph. I create features representing the intrasentential discourse relations found in the current sentence, as well as the intersentential discourse relations connecting the current sentence with the previous one. Each discourse relation produced by the parser yields a feature for its discourse relation type:

$< DiscRelType >$

I also create features designed to (approximately) recognize shared discourse focus. I consider the noun phrases in three syntactic positions: subject, direct object, and the objects of "by" prepositional phrases (PP-by). Sentences in active voice constructions are typically focused on the entities in the subject and direct object positions as the central entities of the discourse. Sentences in passive voice constructions are usually focused on the entities in the subject and PP-by positions as the most central entities. I use the Stanford parser [68] to identify these syntactic constituents.

The motivation for this type of feature is that sentences which have a shared discourse focus probably should be assigned the same event label (i.e., if one of the sentences is discussing a domain-relevant event, then the other probably is too). To capture the intuition behind this idea, consider the following two sentences:

*(1) A customer in the store was shot by masked men.*
*(2) The two men used 9mm semi-automatic pistols.*

Because the same entity (the men) appears in both the "by" PP of sentence (1) and the subject position of sentence (2), the classifier should recognize that the second sentence is connected to the first. Recognizing this connection may enable the extraction system to correctly identify the pistols as instruments used in the shooting event, even though sentence (2) does not explicitly mention the shooting.

I create a discourse focus feature for each shared noun phrase that occurs in two adjacent sentences in one of the designated syntactic positions. I consider any two noun phrases that have the same head word to match. I encode each feature as a triple consisting of the head word of the shared noun phrase ($NPHead$), the NP's position in the current sentence ($SynPos_i$), and the NP's position in the preceding sentence ($SynPos_{i-1}$):

$< NPHead, SynPos_i, SynPos_{i-1} >$

For example, sentences (1) and (2) would produce the following discourse focus feature:

$< men, subject, PP\text{-}by >$

### 4.2.4 Role Filler Distribution Features

The motivation for the bottom-up design of our event extraction architecture is that the sentence classifier can benefit from knowledge of probable role fillers hypothesized by the local detectors. Intuitively, the presence of multiple role fillers within a sentence or in the preceding sentence is a strong indication that a domain-relevant event is being discussed. The local detectors are not perfect, but they provide valuable clues about the number, types, and density of probable role fillers in a region of text.

First, I create features that capture information about the candidate role fillers within a single sentence. I create features for the event role type and the head noun of each candidate role filler in the sentence. I also encode two types of features that capture properties of the set of candidate role fillers. For each event role, I define a binary feature that indicates whether there are multiple candidate role fillers for that role. For example, if we see multiple victims in a sentence, this is more evidence than seeing a single victim. The second type of feature represents combinations of different event role types detected in the same sentence. I define binary features that represent the presence of pairs of distinct event roles occurring in the same sentence.[2] For example, if we see both a candidate perpetrator and a candidate victim in a sentence, we may be more confident that the sentence is describing a crime.

I also create several types of features that represent role filler distributions across sentences. Intuitively, the presence of a particular type of role filler in one sentence may predict the presence of a role filler in the next sentence. For example, a gun is more likely to be an instrument used in a crime if the preceding sentences mention perpetrators and victims than if they only mention other weapons. To capture domain-specific distributional properties of the candidate role fillers, I create features for the role fillers found in adjacent sentences. I use both the head word of the noun phrase as well as the type of the event role. If the local detectors produce a candidate role filler of type $RFType_{i-1}$ with head $RFHead_{i-1}$ in the previous sentence, and a role filler of type $RFType_i$ with head $RFHead_i$ in the current sentence, then two features are generated:

$< RFHead_{i-1}, RFType_i >$

$< RFHead_{i-1}, RFType_{i-1}, RFType_i >$

[2]If there are 5 event roles, there are 10 pairs of distinct roles because the order of them does not matter.

For example, assuming that three candidate role fillers have been detected for the example sentences in Section 4.2.3 (*Victim(customer)* and *Perpetrator(men)* from sentence (1) and *Weapon(pistols)* from sentence (2)), the following features will be created:

$< customer, Weapon >$

$< customer, Victim, Weapon >$

$< men, Weapon >$

$< men, Perpetrator, Weapon >$

I also create features to represent role fillers that occur in adjacent sentences and share a discourse relation. If two adjacent sentences share a discourse relation ($DiscRelType$), then I represent the types of role fillers found in those sentences, coupled with the discourse relation. For example, if two sentences are in a causal relation and the candidate role filler detectors found a candidate victim in the previous sentence and a candidate perpetrator in the current sentence, then the causal relation provides further evidence that the victim and perpetrator are likely correct. These types of features are represented as:

$< RFType_{i-1}, DiscRelType, RFType_i >$

For the example above, the feature would be:

$< Victim, cause, Perpetrator >$

Finally, verbs often provide valuable clues that a sentence is discussing an event, so the presence of a specific verb in the previous sentence may bolster a role filler hypothesis in the current sentence. I create an additional feature that links each verb in the previous sentence to each candidate role filler in the current sentence:

$< Verb_{i-1}, RFType_i >$

For example, a sentence containing a candidate victim preceded by a sentence containing the word "bombed" would produce the following feature:

$< bombed, Victim >$

### 4.2.5 System Generated vs. Gold Standard Role Fillers

When generating these features during training, the gold standard role fillers are not suitable because gold role fillers will not be available in new texts. A model trained with gold role fillers would probably not be effective when applied to new documents that have less

reliable system-generated candidate role fillers. To obtain realistic values for the candidate role filler distributions, I used 5-fold cross-validation on the training data. To get the candidate role fillers for one fold, I trained the role filler detectors using the other four folds and then applied the detectors to the selected fold.

## 4.3 Evaluation

Similar to *TIER*, I will evaluate the implemented system on two event domains to show the generality of my unified discourse-guided event extraction architecture. The data sets are the same as the ones used to evaluate *TIER*: the MUC-4 terrorism corpus and the Civil Unrest data set. The MUC-4 terrorism corpus [76] is a standard benchmark collection for evaluating event extraction systems. The Civil Unrest data set is newly annotated for evaluating my discourse-guided event extraction models. Please refer to Section 3.5.1 for details about the two data sets. The evaluation methods (Section 3.5.2) and metrics (Section 3.5.3) are the same as used for evaluating *TIER* too.

## 4.4 Results on the MUC-4 Data Set

In this section, I will show experimental results on the MUC-4 data set. The MUC-4 corpus consists of 1700 documents in total; to be consistent with previously reported results, I use the 1300 DEV documents for training, 200 documents (TST1+TST2) as a tuning set, and 200 documents (TST3+TST4) as the test set.

### 4.4.1 Experimental Results

Table 4.1 shows the evaluation results on the five event roles for the MUC-4 task, and the macro-average over all five roles. Each cell in the table shows the precision (P), recall (R), and F scores, written as P/R/F. The first row of numbers shows the results for the

**Table 4.1**: Experimental Results on the MUC-4 Data Set, Precision/Recall/F-score.

| System | PerpInd | PerpOrg | Target | Victim | Weapon | Average |
|---|---|---|---|---|---|---|
| Local Extraction Only | | | | | | |
| **Candidate RF Detectors** | 25/67/36 | 26/78/39 | 34/83/49 | 32/72/45 | 30/75/43 | 30/75/42 |
| LINKER (with Structured Sentence Classifier) | | | | | | |
| **Basic feature set** | 56/54/55 | 47/46/46 | 55/69/**61** | 61/57/59 | 58/53/56 | 55/56/56 |
| **+ Candidate RF features** | 51/57/54 | 47/47/47 | 54/69/60 | 60/58/59 | 56/60/58 | 54/59/56 |
| **+ Lexical Bridge features** | 51/57/53 | 51/50/50 | 55/69/**61** | 60/58/59 | 62/62/62 | 56/59/57 |
| **+ Discourse features** | 54/57/**56** | 55/49/**51** | 55/68/**61** | 63/59/**61** | 62/64/**63** | 58/60/**59** |
| Previous Systems | | | | | | |
| **TIER (2011)** | 48/57/52 | 46/53/50 | 51/73/60 | 56/60/58 | 53/64/58 | 51/62/56 |
| **GLACIER (2009)** | 51/58/54 | 34/45/38 | 43/72/53 | 55/58/56 | 57/53/55 | 48/57/52 |

candidate role filler detectors when used by themselves. These local role filler extractors produce relatively high recall, but consistently low precision.

The next set of rows in Table 4.1 shows the effect of adding the structured sentence classifier to create the complete bottom-up event extraction model. I incrementally add each set of textual cohesion features to assess the impact of each one separately. The Basic feature set row uses only the N-gram features. Even with just these simple features, incorporating the structured sentence classifier into the model yields a large improvement in precision (+25) but at the expense of substantial recall (-19).

The **+ Candidate RF features** row shows the impact of providing the candidate role filler information to the sentence classifier (see Section 4.2.4). Compared with the previous row, the role filler features produce an average recall gain of +3, with only a 1-point loss of precision. When looking at the event roles individually, we see that recall improves for all of the event roles except Targets.

The **+ Lexical Bridge features** row shows the impact of the lexical bridge features (Section 4.2.2). These features produced a 2-point gain in precision, yielding a 1-point gain in F-score. Two of the event roles (PerpOrg and Weapon) showed improvement in both precision and recall.

The **+ Discourse features** row shows the performance after adding the discourse bridge features (Section 4.2.3). The discourse features improve precision for three of the five event roles (PerpInd, PerpOrg, and Victim). Weapons also gain two points of recall. Overall, the discourse features yield a 2-point increase in the F score.

Together, all of the textual cohesion features yield a 3-point gain in precision and a 4-point gain in recall relative to the basic feature set (N-grams), achieving an F-score improvement of 3 points.

### 4.4.2 Comparison with Other Systems

I compare the performance of the event extraction model *LINKER* with the performance of my first discourse-guided event extraction model *TIER* (Chapter 3). Briefly speaking, TIER is designed to identify secondary role filler contexts in the absence of event keywords by using a document genre classifier, a set of *role-specific sentence classifiers*, one per event role, in addition to an *event sentence classifier* (similar to classifiers used in other work [102, 37]). In TIER's multilayered event extraction architecture, documents pass through a pipeline where they are analyzed at different levels of granularity: document level, sentence level, and phrase level. In addition, I compare *LINKER*'s performance with another relatively recent event extraction system GLACIER [102], which has also been

evaluated on the same MUC-4 data set. GLACIER uses a unified probabilistic model for event extraction that jointly considers sentential evidence and phrasal evidence when extracting each role filler. It consists of an sentential event recognizer and a set of plausible role filler recognizers, one for each role. The final extraction decisions are based on the product of the normalized sentential and the phrasal probabilities.

The last two rows in Table 4.1 show the results for TIER and GLACIER, using the same evaluation criteria as *LINKER*. I compare their results with the performance of *LINKER*'s complete event extraction system using all of the feature sets, which is shown in the + **Discourse Features** row of Table 4.1. Compared with my first discourse-guided event extraction model TIER, *LINKER* achieves 7 points higher precision, although with slightly lower recall (-2). Overall, *LINKER* yields a 3-point higher F score than TIER. If we look at the individual event roles, *LINKER* produces substantially higher precision across all five event roles. Recall is comparable for PerpInd, Victim, and Weapon, but is several points lower on the PerpOrg and Target roles. Compared with GLACIER, *LINKER* also shows significant gains in precision over all five event roles. Furthermore, the average recall is 3 points higher, with Weapons showing the largest benefit (+11 recall gain).

In summary, the unified discourse-guided event extraction model *LINKER* yields substantially higher precision than previous event extraction systems on the MUC-4 data set, with similar levels of recall. Considering the limited number of documents (200 documents) in the test set, in Section 4.6, I will show the statistical significance testing results by comparing *LINKER*'s performance to *TIER*'s performance.

## 4.5 Results on the Civil Unrest Data Set

Compared to the MUC-4 corpus, which consists of 1700 documents, the Civil Unrest data set (372 documents) is much smaller. Therefore, the same as in *TIER* evaluations, on the Civil Unrest data set, I performed 10-fold cross-validation to evaluate *LINKER* too.

### 4.5.1 Experimental Results

To facilitate comparisons, the candidate role filler extractors are exactly the same as used in *TIER*. The first row of Table 4.2 shows the results of using the candidate role filler extractors only. The second section of the table shows the extraction systems' performance with the structured sentence classifier. The **Basic feature set** row shows that using only the basic sentential features, the structured sentence classifier substantially improved the precision. The **+ Candidate RF features** row shows that after adding the domain-specific role filler distributional features, the structured sentence classifier can identify more relevant

**Table 4.2**: Experimental Results on the Civil Unrest Data Set, Precision/Recall/F-score

| System | Agent | Site | Location | Instrument | Average |
|---|---|---|---|---|---|
| Local Extraction Only | | | | | |
| **Candidate RF Detectors** | 37/51/43 | 23/38/28 | 13/49/21 | 44/70/54 | 29/52/38 |
| with Structured Sentence Classifier | | | | | |
| **Basic feature set** | 66/28/40 | 57/19/29 | 61/22/33 | 70/55/62 | 64/31/42 |
| **+ Candidate RF features** | 57/40/47 | 43/27/33 | 40/33/36 | 64/61/63 | 51/40/45 |
| **+ Lexical Bridge features** | 57/39/46 | 45/28/34 | 42/33/37 | 64/57/60 | 52/39/45 |
| **+ Discourse features** | 58/41/48 | 43/27/33 | 41/33/36 | 64/61/63 | 51/40/45 |
| Previous Systems | | | | | |
| **TIER (2011)** | 50/39/44 | 41/28/34 | 30/29/29 | 67/64/66 | 47/40/43 |

contexts, therefore, it achieved better recall compared to the previous row, with sacrifice in precision. The third row **+ Lexical Bridge features** shows that in this domain, the cross-sentence lexical features can improve the precision for two event roles, Site and Location, but the recall was reduced on the other two event roles, Agent and Instrument. Overall, the lexical features are not helping on top of the role filler distributional features. Therefore, I removed the lexical featuers and added in the discourse relation features to check if the cross sentence discourse bridge features can further improve the structured sentence classifier's performance on top of the role filler distributional features. The fourth row **+ Discourse features** shows that the discourse features can mildly further improve the extraction precision for two event roles, Agent and Location, while maintaining the overall recall.

Compared to *TIER*'s results (the last row of the table) on the Civil Unrest data set, *LINKER* achieved much better precision while with some loss of recall, and overall achieved a slightly better F-score. Similar to the evaluations using the MUC-4 data set, the test set (all the data set in the cross-validation setting) of the civil unrest domain is also limited in size; therefore, in Section 4.6, I will also show the statistical significance testing results by comparing *LINKER*ś performance to *TIER*ś performance using the civil unrest data set.

### 4.5.2 Learning Curve

Similar to what we have observed in *TIER*'s evaluations, the performance of *LINKER* is also relatively low in the Civil Unrest domain compared with its evaluation results on the MUC-4 terrorism domain. This is probably due, at least in part, to the smaller amount of training data. To show how *LINKER*'s performance was affected by the size of data, I also drew the learning curve for *LINKER* by training the system on different subsets of the data set. The procedures are the same as used in the learning curve for *TIER* (see Section 3.7.2). From the learning curve shown in Figure 4.4, we can see that the extraction performance

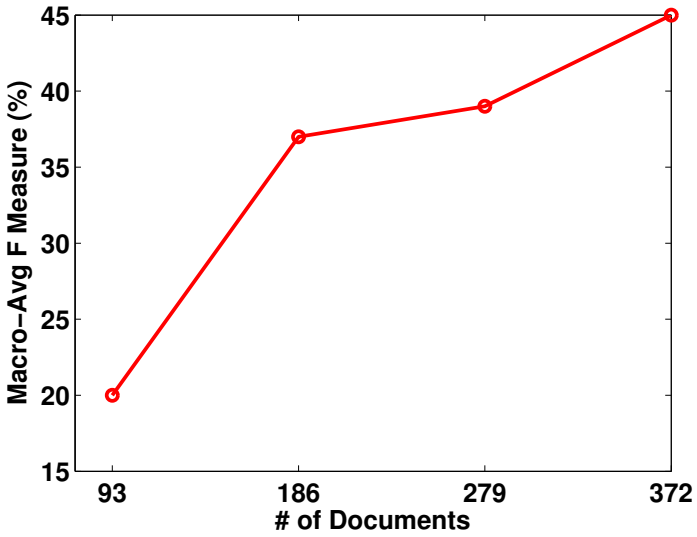


**Figure 4.4**: Learning Curve for LINKER

of *LINKER* clearly improves when more and more training data were used. Therefore, we can expect to see better extraction performance if we train *LINKER* with a larger data set in the future.

## 4.6 Statistical Significance Testing

Up to this point, I have presented the evaluation results of both *TIER* and *LINKER* on two distinct event domains: the MUC-4 terrorism corpus and the newly created Civil Unrest data set. Overall, *LINKER* has achieved better results when compared to its counterpart, *TIER*. Table 4.3 shows the performance comparisons using the macro-average. Considering the imbalance of role filler distributions in both domains, I also calculated the micro-averages and Table 4.4 shows a summary of the comparisons.

To see if *LINKER* has achieved statistically significant extraction performance improvements compared to *TIER*, I ran significance testing using **F-scores** on both data sets. The testing methodology is paired bootstrap [14]. Table 4.5 reports the significance testing results.

We can see that on the terrorism domain (the MUC-4 corpus), *LINKER* performs significantly better than *TIER* using both Macro Average and Micro Average F-scores measurements. On the second event domain (Civil Unrest data set), *LINKER* also performs significantly better than *TIER* at the $p < 0.1$ level (Macro Average) and the $p < 0.01$ level (Micro Average).

**Table 4.3**: Macro-Average Evaluation Summary (Precision/Recall/F-score)

| Domains | TIER | LINKER |
|---|---|---|
| **Terrorism (MUC-4)** | 51/62/56 | 58/60/59 |
| **Civil Unrest (CU)** | 47/40/43 | 51/40/45 |

**Table 4.4**: Micro-Average Evaluation Summary (Precision/Recall/F-score)

| Domains | TIER | LINKER |
|---|---|---|
| **Terrorism (MUC-4)** | 53/61/57 | 58/60/59 |
| **Civil Unrest (CU)** | 46/38/41 | 53/39/45 |

**Table 4.5**: Significance Testing Results ($p$ levels) for LINKER vs. TIER

| Domains | Macro_Avg | Micro_Avg |
|---|---|---|
| **Terrorism (MUC-4)** | 0.05 | 0.05 |
| **Civil Unrest (CU)** | 0.1 | 0.01 |

## 4.7 Remarks on Discourse-Guided Event Extraction

At first glance, *TIER* and *LINKER* are rather different in their architectures. To identify event regions, *TIER* contains three text analysis components that work at both document and sentence levels while *LINKER* used a single structured sentence classifier to examine a whole document. However, these two models are inherently connected and *LINKER* was actually proposed to address limitations of *TIER*, which should be viewed as my first cut discourse-guided event extraction architecture.

It has been a challenging goal to go beyond processing individual sentences for extracting information because there are innumerable ways in which pieces of information are related. *TIER* was designed to indirectly tackle discourse issues in event extraction by progressively "zooming in" on relevant information. Specifically, it distinguishes event narrative documents from fleeting references (i.e., the documents that only briefly mention an event and then switch to other topics) and allows us to more aggressively hunt for event-related information in event narratives. However, there are clearly many discourse phenomena exhibited across sentences within a document; for instance, the use of certain words in a discourse unit (e.g., sentence) tends to trigger the use of other words in subsequent discourse units (i.e., lexical word associations). By using a rich discourse-oriented feature set in a structured learning algorithm, *LINKER* can explicitly capture discourse continuity across sentences and model a diverse set of discourse issues in a single structured event sentence recognizer which captures interactions of multiple discourse phenomena and their collective influences to event story telling.

Empirical evaluation on two event domains has shown that *LINKER* performed significantly better than *TIER*. By design, *LINKER* does improve on *TIER*. However, for fleeting reference documents, it might be a waste using the sophisticated *LINKER* to extensively model various discourse links and search for event-relevant sentences in those documents. We could potentially combine the two models to create a more efficient extraction system by maintaining the two-pronged architecture of *TIER* and bringing in the structured event sentence recognizer of *LINKER* when processing event narrative documents. In addition, from a practical perspective, *TIER* is well modularized and its components can be easily trained in an independent manner; therefore, in practice, *TIER* might be more robust than *LINKER*, especially when only a limited amount of annotated data is available. In contrast, while *LINKER* is powerful enough to model a variety of discourse phenomena across sentences, more labeled documents might be needed to train a system to perform well.

Overall, on two distinct event domains, both of my discourse-guided event extraction models have produced better extraction performance compared to previous systems. These results show that modeling discourse phenomena can be beneficial for event extraction. However, we should be cautious to draw the conclusion that the presented models will improve event extraction performance uniformly in all scenarios. We should realize that each of the two data sets I experimented on is a collection of news articles and formal news reports are generally characteristic of coherent discourse flows. It is still a question if discourse-guided event extraction architectures will be useful when processing texts of different genres, for instance, short informal user-generated texts in social media. But in general, discourse-guided models should be beneficial for event extraction from news articles.

# CHAPTER 5

# MULTIFACETED EVENT RECOGNITION

Before giving documents to sophisticated event extraction systems, we want to ask if the documents actually contain any relevant events, mainly for two reasons. First, because event extraction generally comprises costly text analysis, processing documents that do not mention a relevant event is a waste of computing resources. Second, by focusing on event-relevant documents, event extraction systems can be more accurate because any extraction that is produced from irrelevant documents is a false hit and makes event extraction less precise. In this chapter, I will present my research on *event recognition* which aims to accurately identify documents that describe a specific type of event.

*Event recognition* can facilitate a series of easily scalable event-oriented applications. One example is tracking events. Many people are interested in following news reports and updates about events. Government agencies are keenly interested in news about civil unrest, acts of terrorism, and disease outbreaks. Companies want to stay on top of news about corporate acquisitions, high-level management changes, and new joint ventures. The general public is interested in articles about crime, natural disasters, and plane crashes. With accurate event recognition, we can detect the first occurrences and the following mentions of particular types of events; thus, we can track the dynamics of events.

## 5.1 Challenges to Accurate Event Recognition

Event recognition is a challenging task. It is tempting to assume that event keywords are sufficient to identify documents that discuss instances of an event, but event words are rarely reliable on their own. For example, consider the challenge of finding documents about civil unrest. The words *"strike"*, *"rally"*, and *"riot"* refer to common types of civil unrest, but they frequently refer to other things as well. A strike can refer to a military event or a sporting event (e.g., *"air strike"*, *"bowling strike"*), a rally can be a race or a spirited exchange (e.g.,*"car rally"*, *"tennis rally"*), and a riot can refer to something funny (e.g., *"she's a riot"*). Event keywords also appear in general discussions that do not mention a specific event (e.g., *"37 states prohibit teacher strikes"* or *"The fine for inciting a riot is*

*$1,000"*). Furthermore, many relevant documents are not easy to recognize because events can be described with complex expressions that do not include event keywords. For example, *"took to the streets"*, *"walked off their jobs"*, and *"stormed parliament"* often describe civil unrest.

## 5.2 Event Facets: To the Rescue

While event expressions are not sufficient to unambiguously recognize event descriptions of a particular type, events generally feature certain characteristics that are essential to distinguish one type of event from another. I call the defining characteristics of an event *"event facets"*. For example, *agents* and *purpose* are event facets for many types of events.

The agent responsible for an action often determines how we categorize the action. For example, natural disasters, military operations, and terrorist attacks can all produce human casualties and physical destruction. However, the agent of a natural disaster must be a natural force, the agent of a military incident must be military personnel, and the agent of a terrorist attack is never a natural force and rarely military personnel. Therefore, the agent is often an essential part of an event definition.

The purpose of an event is also a crucial factor in distinguishing between some event types. For example, civil unrest events and sporting events both involve large groups of people amassing at a specific site. However, the purpose of civil unrest gatherings is to protest against socio-political problems, while sporting events are intended as entertainment. As another example, terrorist events and military incidents can both cause casualties, but the purpose of terrorism is to cause widespread fear, while the purpose of military actions is to protect national security interests.

In addition to agents and purposes, there are other event facets, such as cause of events, effects, and patients of events, which are necessary to distinguish many types of events too. The cause is analogous to the agent of an event, except that causes can refer to natural forces or activities that lead to events while agents generally refer to humans that initiate events. Similarly, the effect is closely related to the purpose of an event because commonly effects serve as a reflection of purposes. However, their emphases are different. Effects stress the consequences resulting from events while purposes of events point to the original motivations promoting events. Another common defining characteristic is the entity or object affected by an event, which I will refer to as the "patient". For example, by definition, a plane crash event must involve a plane that crashed, and a corporate acquisition must involve a company that was acquired. Depending on the types of events, there can be other important factors as well, which are key to define events.

An interesting angle to organize a variety of events in both nature and our social lives is by using the types of their event-defining characteristics. Therefore, seemingly unrelated events can be put into a general event group because they share the same types of event facets. For example, conceptually, vehicle crashes and corporate acquisitions are two completely different types of events. However, vehicle crashes must involve vehicles that are crashed and corporate acquisitions will always refer to the corporate entities that are acquired. Therefore, structurally, these two types of events feature the same event facet, patients of events, and they naturally fall into a general event group. Research in event ontology across multiple related disciplines, such as philosophy, cognition, or knowledge engineering ([36, 115, 53]), have shown similar observations. This provides a perspective to validate that event defining characteristics are necessary complements, in addition to event expressions, for recognizing events.

Table 5.1 lists 16 event domains that I organize based on the event facets they have. Many of the event domains are from community-wide performance evaluations, including Message Understanding Conferences (MUCs) and Automatic Content Extraction (ACE) evaluations. Benefitting from the availability of human annotations for event recognition or extraction purposes, events such as terrorism, management successions, and disease outbreaks have been extensively studied in the community.

Roughly, Table 5.1 groups 16 event types into 6 categories. The first group of events shares two event facets, agents and purpose. By definition, civil unrest is a broad term that is typically used by the media or law enforcement to describe a form of public disturbance caused by a group of people for a purpose. Civil unrest requires a population group to participate in unrest events and a specific purpose that motivates the events. Similarly, referendums feature event agents and purpose too.

*Effects* (consequences) of events are essential to large-scale influential events because effects generally reflect impacts of the events and explain why the events are newsworthy. Therefore, the second group of events, including terrorism events, military operations and shark attacks, have agents as well as effects as facets. For example, people care about terrorism events partly because of the massive human casualties or physical losses that often result from the terrorism (effects). Similarly, we expect to see people injured in shark attacks too. It is surprising if effects are not mentioned in these events. In addition, the event facet *patient* is also important to define events in this group. We can see that each of the three types of events is characteristic of certain kinds of patients. For instance, terrorism events tend to have civilians as the target while military operations generally

**Table 5.1**: Grouping of Event Types Based on Event Facets

| | Agent | Patient | Purpose | Effect | Cause | Medium |
|---|---|---|---|---|---|---|
| **Civil Unrest** | ✓ | | ✓ | | | |
| **Referendum** | ✓ | | ✓ | | | |
| **Terrorism** | ✓ | ✓ | | ✓ | | |
| **Military Operations** | ✓ | ✓ | | ✓ | | |
| **Shark Attacks** | ✓ | ✓ | | ✓ | | |
| **Natural Disasters** | | | | ✓ | ✓ | |
| **Disease Outbreaks** | | | | ✓ | ✓ | |
| **Vehicle Crash** | | ✓ | | | | |
| **Rocket/Missile Launches** | | ✓ | | | | |
| **Corporate Acquisition** | | ✓ | | | | |
| **Microelectronic Production** | | ✓ | | | | |
| **Management Succession** | ✓ | ✓ | | | | ✓ |
| **Fine** | ✓ | ✓ | | | | ✓ |
| **Negotiation** | ✓✓ | | | | | |
| **Sports Games** | ✓✓ | | | | | |
| **Meeting** | ✓✓ | | | | | |

involve soldiers that were killed during the combats. More interestingly, we expect to see humans wounded or killed in shark attacks, but we seldom refer to the act of a shark killing another fish as a "shark attack", especially in news reports.

The third group of events are not initiated by people, but by a natural cause. The examples of this category include natural disasters and disease outbreaks. For similar reasons as in the second group, the facet *effect* is also necessary to define events in this group. Essentially, we pay attention to natural disasters and disease outbreaks because of the significant influences these events can make to our living environment or other aspects of life.

In my event facet analysis, I found that the *patient* of events is the only defining characteristic for several types of events. The fourth group covers specific event types such as vehicle crashes and corporate acquisitions. These event types seem distinct from each

other; however, across all these events, the patients are central in their event definitions. For example, some vehicle must have been crashed in vehicle crash events while a corporate acquisition event report is expected to refer to the company that is acquired. In contrast, different from the formerly discussed event groups, agents or causes are not so vital for events in this group because without realizing who or what caused the events, we can still recognize these events once we know certain types of patients that define the corresponding events.

There are also events that involve a transfer of certain physical object or abstract concept from one party to the other. I will name the transferred object or concept as *medium*, the party that acquires the medium as agents, and the party that releases the medium as patients of events. Management successions and monetary fines fall into this category. Respectively, the medium refers to the position that changes in a management succession or the monetary amount that changes hands in a fine. I also identify one category of event that involves two agent parties. For example, negotiations generally involve two agents speaking, similarly for meetings. In sports games or other competitions, two agent parties are essentially the main factors of events.

Through the above analysis of facets in well-studied event domains, we can see that event facets can effectively reveal event structures and are crucial to define events. Therefore, event facets are important for accurately identifying events. However, we should realize that neither the event facet types nor the event groups listed in Table 5.1 are complete. Depending on the new types of events, we probably need new event facets to define them. Rather, the event analysis shown in Table 5.1 is meant to take the event types that have been relatively well studied as examples and explain how event facets are necessary to define events. Hopefully, the demonstrated event grouping will provide useful guidance on how we identify event facets for any given type of events.

## 5.3 Multifaceted Event Recognition

My research explores the idea of *multifaceted event recognition*: using event expressions as well as facets of the event to identify documents about a specific type of event. In addition to event expressions, event facets can provide valuable complementary information to accurately detect documents describing a particular type of events. For example, as illustrated previously, using the event keyword "rally" alone will not unambiguously recognize civil unrest event descriptions because "rally" is frequently used to refer to other types of events too, e.g., *"car rally"*, *"tennis rally"*. However, if we know the agents or purpose of an

event, we can better picture the event scenario and determine what type of event is being discussed in text. For example, in addition to the event keyword "rally", knowing that the event participants (agents) are "coal miners" or the goal of the rally is to "press for higher wages", we immediately gain confidence that some civil unrest event is being referred to. Event facet information is so valuable that observing multiple types of event facets in text can sometimes suggest a particular type of event event without an event phrase. For example, without seeing any explicit event expression, if both a plausible civil unrest agent (e.g., "coal miners") and a plausible civil unrest purpose (e.g., "press for higher wages") are mentioned in the context, then we may hypothesize that a civil unrest event is being discussed.

## 5.4 Bootstrapped Learning Framework of Event Dictionaries

I present a bootstrapping framework to automatically learn event phrase and event facet dictionaries. The learning process uses unannotated texts and minimal human supervision that includes a few event keywords and seed terms for each type of event facet associated with the event type. My bootstrapping algorithm exploits the observation that event expressions and event facets often appear together in text regions that introduce an event. Furthermore, seeing multiple types of event information in a localized text region often implies that a relevant event is being described and we can look for additional types of event information within the same text region. Based on these observations, I designed a bootstrapping algorithm that ricochets back and forth, alternately learning new event phrases and learning new event facet phrases in an iterative process.

Specifically, each learning iteration of the bootstrapping algorithm consists of two learning stages. The first stage is designed to learn event phrases while the second stage is to learn event facet phrases. In the following sections, I will elaborate with more details on how the two learning stages proceed.

### 5.4.1 Stage 1: Event Phrase Learning

The learning process will start with unannotated texts. Because a particular type of event, for example civil unrest events or plane crashes, does not happen constantly and therefore is relatively infrequent in a broad coverage collection of news reports, a small number of event keywords can be used to create a pseudo domain specific corpus by requiring each document in the corpus to contain at least one event keyword. However, as explained previously, event keywords are not sufficient to obtain relevant documents with

high precision, so the extracted stories are a mix of relevant and irrelevant articles. My algorithm first selects text regions to use for learning, and then harvests event expressions from them.

#### 5.4.1.1 Event Region Identification

Event facets, as defining characteristics of events, can effectively be used to recognize events of a particular type. Seeing multiple types of event facet information together implies that the text region probably is describing a relevant event. For example, if we see the agent "terrorists", the patient "the city mayor", and the effect "was shot to death" in a localized text region, then we are almost certain that some terrorism event is being described.

I identify probable event regions as text snippets that contain at least one phrase of each type of defining facet for the event. To initiate the bootstrapping algorithm, I will identify probable event regions using the seed terms for each type of event facet. As the learning proceeds, new facet terms will be learned and used to enrich facet dictionaries, and more event regions will be identified.

#### 5.4.1.2 Harvesting Event Expressions

Although event expressions and event facet information can appear in text as a variety of complex forms (can be whole sentences for example), to constrain the learning process, I require both event phrase and event facet expressions to match certain syntactic forms. Additionally, I require predefined dependency relations between event expression and event facets, and between pairs of event facets if needed. The dependency constraints will further purify the learning process to control the quality of learned event expressions and event facet phrases. Only event expressions that match the defined syntactic forms and occur in the dependency relations with facet phrases will be extracted.

Naturally, both types of syntactic constraints depend on the set of event facets that characterize a specific type of event. When some event facets become different and accordingly their semantics, syntactic forms used to identify individual event expressions and event facet candidates as well as dependency relations among them can be different too.

### 5.4.2 Stage 2: Event Facet Phrase Learning

Similarly, the stage for event facet learning also consists of two steps, event region identification and event facet phrase extraction. However, different from the first stage where only one learning process goes on to learn event expressions, multiple learning

processes are active in this stage, one per event facet. Each learning process is to learn phrases of a particular type of event facet and they will proceed in parallel.

#### 5.4.2.1 Event Region Identification

To identify event regions for learning phrases of a particular event facet, I use the event expressions and event facet phrases of all types except the event facet type that is to be learned. Specifically, the text regions that contain at least one event expression and one facet phrase of all the other types are selected. The event expressions and event facet phrases are from the dictionaries that have been learned up to this point.

#### 5.4.2.2 Harvesting Event Facet Phrases

Similar to event phrase learning, only event facet candidates that match certain syntactic forms and occur in the predefined dependency structures will be extracted.

As mentioned previously, both types of syntactic learning constraints are dependent on set of facets. Given a particular type of event, we need to first identify its set of event facets, then according to their semantics, certain syntactic forms need to be considered to learn event facet phrases. Furthermore, dependency relations between an event facet and event expressions, and dependency relations between different types of event facets also need to be modified to reflect their specific relations in a particular type of events.

### 5.4.3 Defining Syntactic Forms to Harvest Event Facets

The event facets that I have identified fall into two classes. The first class covers the entities that participate or are involved in events, such as agents and patients of events. The other class covers states, goals and actions that are in certain relations with the target event, such as effects/consequences and reasons/purposes of events. The event facets that fall into the first class are generally noun phrases when appearing in text and syntactically, they can be subjects of verbs, objects of verbs, or objects of prepositions. For example, agents are generally subjects of event phrases while patients are commonly seen as direct objects in event phrases. In contrast, the event facets in the second class, such as purposes, can be formed as verb phrases or simple clausal forms. For example, to learn effects together with patients of terrorism events, simple verb phrases in active voice may capture many effect realizations (e.g., “destroyed an office building” in terrorism events); effects can be seen as verb phrases in passive voice too, due to its semantics, for example, “many houses were damaged”. In addition, effects commonly occur in text as prepositional phrases, for example, “murder of the President” and “killing of university students”.

### 5.4.4 Linking Event Facets to Event Expressions

The general observation that guides the bootstrapping algorithm is that event facet information and event expressions tend to occur close together in event descriptions. In addition, to constrain the learning process, certain dependency relations should be required between event expressions and an event facet, and between different types of event facets. However, the closeness that is necessary for a successful learning process varies depending on the event types, similarly for the dependency relations. For example, in civil unrest event descriptions, I have frequently seen event introductory sentences that contain an event expression, an agent, as well as the purpose of the event. However, I have also observed that in terrorism event descriptions, while event expressions and agents are often coupled, they occur occur in different sentences from the sentence that describes the patient and effect information.

## 5.5 Bootstrapped Learning of Event Dictionaries

In this section, I will describe specific syntactic constructions and dependency structures that are needed to constrain the learning of both event expressions and event facet information in two concrete domains: civil unrest events and terrorism events.

### 5.5.1 Learning Dictionaries for the Civil Unrest Event Domain

For civil unrest events, I have identified agents and purpose as two event facets. Overall, my bootstrapping approach consists of two stages of learning, as shown in Figure 5.1. The process begins with a few agent seeds, purpose phrase patterns, and unannotated articles selected from a broad-coverage corpus using event keywords. In the first stage, event expressions are harvested from the sentences that have both an agent and a purpose phrase in specific syntactic positions. In the second stage, new purpose phrases are harvested from sentences that contain both an event phrase and an agent, while new agent terms are harvested from sentences that contain both an event phrase and a purpose phrase. The new terms are added to growing event dictionaries, and the bootstrapping process repeats.

I first extract potential civil unrest stories from the English Gigaword corpus [83] using six civil unrest keywords. The event keywords include “protest”, “strike”, “march”, “rally”, “riot”, and “occupy”, or their grammatical variations. The input in stage 1 consists of a few agent terms and purpose patterns for seeding. The agent seeds are single nouns, while the purpose patterns are verbs in infinitive or present participle forms. Table 5.2 shows the agent terms and purpose phrases used in the experiments. The agent terms were manually selected by inspecting the most frequent nouns in the documents with civil unrest keywords.

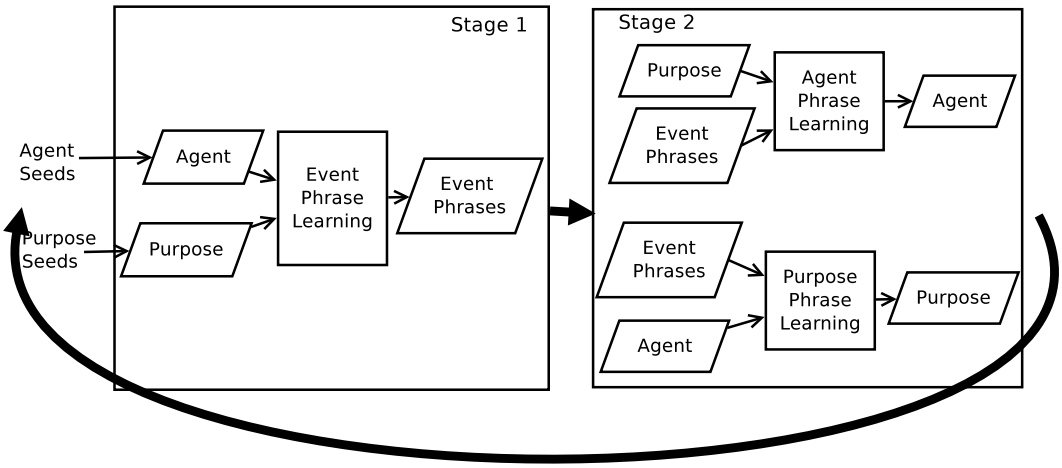


**Figure 5.1**: Bootstrapped Learning of Event Phrases and Event Facet Phrases for Civil Unrest Event Domain.

**Table 5.2**: Agent and Purpose Phrases Used as Seeds in the Civil Unrest Domain

| | |
|---|---|
| **Agents** | protesters, activists, demonstrators, students, groups, crowd, workers, palestinians, supporters, women |
| **Purpose Phrases** | demanding, to demand, protesting, to protest |

The purpose patterns are the most common verbs that describe the reason for a civil unrest event.

As explained previously, to constrain the learning process, I require event expressions and purpose phrases to match certain syntactic forms. I apply the Stanford dependency parser [68] to the probable event sentences, which contain at least one phrase from each event facet, to identify verb phrase candidates and to enforce syntactic constraints between the different types of event information.

#### 5.5.1.1 Syntactic Forms

For our purposes, we learn agent terms that are single nouns; specifically, they are heads of noun phrases. Both event phrases and purpose phrases are verb phrases. Figure 5.2 shows the two types of verb phrases that the system learns. One type consists of a verb paired with the head noun of its direct object (dobj). For example, event phrases can be *"stopped work"* or *"occupied offices"*, and purpose phrases can be *"show support"* or *"condemn war"*. The second type consists of a verb and an attached prepositional phrase, retaining only the head noun of the embedded noun phrase. For example, *"took to street"* and *"scuffled with police"* can be event phrases, while *"call for resignation"* and *"press for wages"* can be

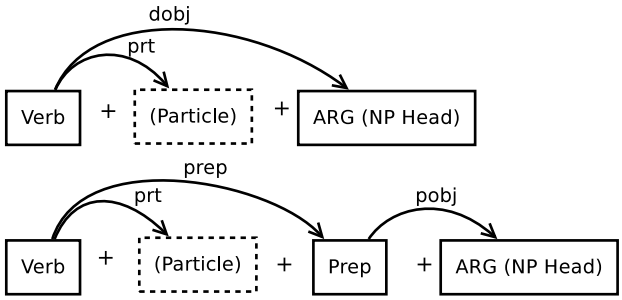


**Figure 5.2**: Phrasal Forms of Event and Purpose Phrases for Civil Unrest Events

purpose phrases. In both types of verb phrases, a particle can optionally follow the verb.

#### 5.5.1.2 Dependency Relations

Event expressions, agents, and purpose phrases must appear in specific dependency relations, as illustrated in Figure 5.3. An agent must be the syntactic subject of the event phrase. A purpose phrase must be a complement of the event phrase; specifically, I require a particular dependency relation, "xcomp", between the two verb phrases. In the dependency parser, "xcomp" denotes a general relation between a VP or an ADJP and its open clausal complement. For example, in the sentence *"He says that you like to swim."*, the "xcomp" relation will link "like" (head) and "swim" (dependent). With my constraints on the verb phrase forms, the dependent verb phrase in this construction tends to describe the purpose of the verb phrase. For example, in the sentence *"Leftist activists took to the streets in the Nepali capital Wednesday protesting higher fuel prices."*, the dependency relation "xcomp" links *"took to the streets"* with *"protesting higher fuel prices"*.

Given the syntactic dependencies shown in Figure 5.3, with a known agent and purpose phrase, I extract the head verb phrase of the "xcomp" dependency relation as an event phrase candidate. The event phrases that co-occur with at least two unique agent terms and two unique purposes phrases are saved in our event phrase dictionary.

The sentences that contain an event phrase and an agent are used to harvest more purpose phrases, while the sentences that contain an event phrase and a purpose phrase are

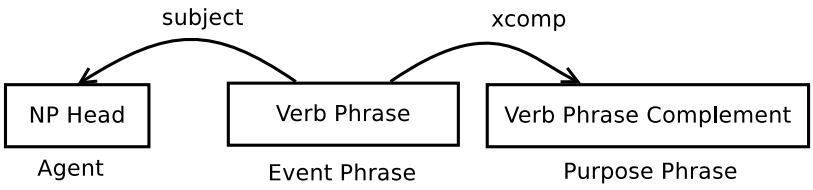


**Figure 5.3**: Syntactic Dependencies between Agents, Event Phrases, and Purpose Phrases

used to harvest more agent terms. Purpose phrases are extracted from the phrasal forms shown in Figure 5.2. In the learning process for agents, if a sentence has an event phrase as the head of the "xcomp" dependency relation and a purpose phrase as the dependent clause of the "xcomp" dependency relation, then the head noun of the syntactic subject of the event phrase is harvested as a candidate agent term. I also record the modifiers appearing in all of the noun phrases headed by an agent term. Agent candidates that co-occur with at least two unique event phrases and at least two different modifiers of known agent terms are selected as new agent terms.

The learning process for purpose phrases is analogous. If the syntactic subject of an event phrase is an agent and the event phrase is the head of the "xcomp" dependency relation, then the dependent clause of the "xcomp" dependency relation is harvested as a candidate purpose phrase. Purpose phrase candidates that co-occur with at least two different event phrases are selected as purpose phrases.

The bootstrapping process then repeats, ricocheting back and forth between learning event phrases and learning agent and purpose phrases.

#### 5.5.1.3 Domain Relevance Criteria

Because the unannotated data that are used for learning civil unrest event dictionaries come from the broad coverage corpus Gigaword [83], even after keyword filtering of the documents, the data are still quite noisy. To avoid domain drift during bootstrapping, I use two additional criteria to discard phrases that are not necessarily associated with the domain.

For each event phrase and purpose phrase, I estimate its *domain-specificity* as the ratio of its prevalence in domain-specific texts compared to broad-coverage texts. The goal is to discard phrases that are common across many types of documents, and therefore not specific to the domain. I define the domain-specificity of phrase $p$ as:

$$\text{domain-specificity(p)} = \frac{frequency\ of\ p\ in\ domain{-}specific\ corpus}{frequency\ of\ p\ in\ broad{-}coverage\ corpus}$$

I randomly sampled 10% of the Gigaword texts that contain a civil unrest event keyword to create the "domain-specific" corpus, and randomly sampled 10% of the remaining Gigaword texts to create the "broad-coverage" corpus.[1] Keyword-based sampling is an approximation to domain-relevance, but gives us a general idea about the prevalance of a phrase in different types of texts.

[1]The random sampling was simply for efficiency reasons.

For agent terms, our goal is to identify people who participate as agents of civil unrest events. Other types of people may be commonly mentioned in civil unrest stories too, as peripheral characters. For example, police may provide security and reporters may provide media coverage of an event, but they are not the agents of the event. I estimate the *event-specificity* of each agent term as the ratio of the phrase's prevalence in event sentences compared to all the sentences in the domain-specific corpus. I define an event sentence as one that contains both a learned event phrase and a purpose phrase, based on the dictionaries at that point in time. Therefore, the number of event sentences increases as the bootstrapped dictionaries grow. I define the event-specificity of phrase $p$ as:

$$\text{event-specificity(p)} = \frac{frequency\ of\ p\ in\ event\ sentences}{frequency\ of\ p\ in\ all\ sentences}$$

In my experiments, I required event and purpose phrases to have *domain-specificity* $\geq$ .33 and agent terms to have *event-specificity* $\geq$ .01. The latter value is so small because I simply want to filter phrases that virtually never occur in the event sentences, and I can recognize very few event sentences in the early stages of bootstrapping.

### 5.5.2 Learning Dictionaries for the Terrorism Event Domain

For terrorism events, I have identified agents, patients, and effects of patients as the event facets. Similarly, my bootstrapping approach consists of two stages of learning as shown in Figure 5.4. The process begins with a few seeds for each type of event facet. In addition, the learning process uses the training documents in the MUC-4 corpus, but the annotated labels will not be used. The Gigaword corpus is not so helpful to provide in-domain documents for this terrorism domain, because the evaluation data of MUC-4 is specific to terrorism events that happened in a specific time period (about 20 years ago), and in several specific countries in Latin America. In the first stage, event expressions are harvested from the text regions that have at least one term of each type of event facet. In the second stage, new facet phrases are harvested from text regions that contain both an event phrase and phrases of the other types of event facets. The newly learned phrases are added to the growing event dictionaries, and the bootstrapping process repeats.

Table 5.3 shows the seed terms that are used as input in the first stage of learning. Both the agent seeds and the patient seeds are single nouns, while the effect patterns are verbs in active or passive voices.

In the following section, I will describe the specific syntactic forms that are used to capture event and facet phrases, and the dependency relations between them.

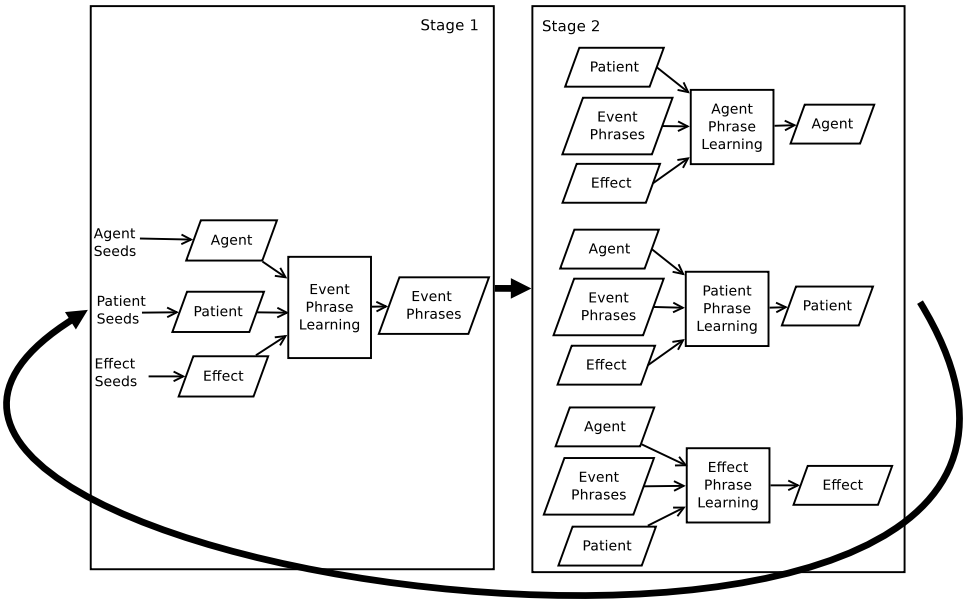


**Figure 5.4**: Bootstrapped Learning of Event Phrases and Event Facet Phrases for the Terrorism Event Domain.

**Table 5.3**: Agent, Patient, and Effect Phrases Used as Seeds in the Terrorism Domain

| | |
|---|---|
| **Agents** | FMLN, Front, ELN, Cartel, Farc, Mrta, squads, guerrillas, terrorists, criminals, rebels, members, individuals, assassins |
| **Patients** | civilians, victims, priests, jesuits, students, women, children, vehicles, offices, residence, building, car, homes, houses, pipeline |
| **Effects** | * be damaged, destroyed *, bombed *, * be murdered, attacked * |

#### 5.5.2.1 Syntactic Forms

The same as in civil unrest events, agents of terrorism events, including the terrorist individuals (e.g., "terrorists") and terrorism organizations (e.g., "FMLN"), will also be defined as single nouns. In addition, patients of terrorism events are also single nouns. Patients of terrorism events include the human targets, such as political leaders that are assassinated, and physical targets, e.g., civilian facilities that are bombed. Event expressions have to be in the syntactic forms as shown in Figure 5.2; this is the same as civil unrest events too. For example, event phrases can be "hit helicopters" or "carried out attacks".

By definition, effects of terrorism events are consequences that happen to patients during or after terrorism activities. Therefore, I require that effects of terrorism events are always coupled with patients; specifically, patients are arguments of effect phrases. Multiple syntactic forms are used to identify effects together with patients.

First, the phrasal forms as shown in Figure 5.2 are also used to identify effect phrases. For example, effect phrases can be "wounded *" or "broke into *". Note that * refers to a patient. In addition, due to its semantics, effects are often described as verb phrases in passive voice or as prepositional phrases headed by event nouns. Therefore, to well capture the diversity of effect expressions in terrorism events, I add three new phrasal forms (as shown in Figure 5.5). The top one identifies effect verb phrases in passive voice. For example, effect phrases can be "* be shot" or "* be destroyed". The later two new phrasal forms (the bottom two) capture effect phrases that occur in text as possessive forms or prepositional phrases headed by nouns. For example, effect phrases can be " *'s murder " (in the first case) or " death of * " (in the second case).

#### 5.5.2.2 Dependency Relations

Different from the civil unrest event domain where event expressions, agents, and purpose phrases must appear in the same sentence, in the terrorism domain, event phrases and the three event facets, agents, patients, and effects of patients do not have to appear in the same sentence. However, following the key observation that event expressions and event facet information should co-occur in localized text regions, I require that terrorism event phrases and the three types of facet information appear together in text segments that span a small number of sentences. Specifically, in my experiments, I require them to occur together within at most four sentences. In other words, the last sentence that contains a piece of event information should be within three sentences from the first sentence that contains other pieces of event information.

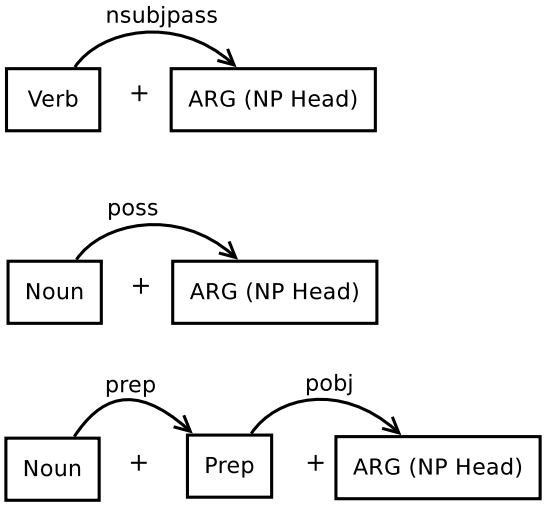


**Figure 5.5**: Three New Syntactic Structures to Extract Effect Phrases and Patient Terms

Furthermore, as described earlier, patients of terrorism events must occur in the same sentence as effects and strictly, patients must be the arguments of effect patterns. In addition, as shown in Figure 5.6, agents of terrorism events must appear in texts as the syntactic subjects of event phrases.

Therefore, in the first learning stage, to learn event phrases, each candidate must have an agent as its syntactic subject. In addition, there has to be a sentence that contains an effect phrase with a patient as its argument; furthermore, the sentence must be within three sentences from the sentence that contains the event phrase candidate. Similarly, in the second learning stage, to learn an event facet phrase, an event phrase and a phrase of the other two types of facets must be seen within a text chunk of at most four sentences, at the same time, the dependency relations must be satisfied; specifically, agents must be the syntactic subject of an event phrase and patients must be the argument of an effect phrase.

## 5.6 Evaluation Design

### 5.6.1 Data

Similar to the evaluation of the two discourse-guided event extraction architectures (*TIER* in Chapter 3 and *LINKER* in Chapter 4), I will evaluate my multifaceted event recognition approach on two distinct event domains. This will verify the general applicability of the multifaceted event recognition approach that aims to accurately identify documents describing a particular type of event. Specifically, I will evaluate using the same two event data sets: the civil unrest event data set and the MUC-4 terrorism corpus. I will create systems that learn event dictionaries and evaluate the performance of the multifaceted dictionary lookup approach to *recognize documents that mention relevant events.*

#### 5.6.1.1 Civil Unrest Event Domain

To refresh, civil unrest is a broad term typically used by the media or law enforcement to describe a form of public disturbance that involves a group of people, usually to protest or promote a cause. Civil unrest events include strikes, protests, occupations, rallies, and

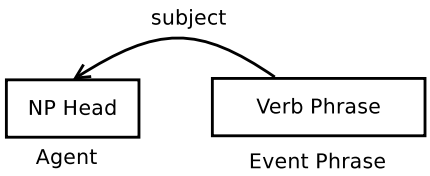


**Figure 5.6**: Syntactic Dependencies between Agents and Event Phrases in Terrorism Domain

similar forms of obstructions or riots. The data set consists of 400 documents and they are annotated as specified in Section 3.5.1. For event recognition evaluation purposes, as a reminder, I will briefly restate how the data set was created.

I chose six *event keywords* to identify potential civil unrest documents: "protest", "strike", "march", "rally", "riot", and "occupy". I extracted documents from the English Gigaword corpus [83] that contain at least one of these event keywords, or a morphological variant of a keyword.[2] This process extracted nearly one million documents, which I will refer to as the *event-keyword corpus*. I then randomly sampled 400 documents[3] from the event-keyword corpus and asked two annotators to determine whether each document mentioned a civil unrest event. I defined annotation guidelines and conducted an interannotator agreement study on 100 of these documents. The annotators achieved a $\kappa$ score of .82. I used these 100 documents as our *tuning set*. Then, each annotator annotated 150 more documents to create our *test set* of 300 documents.

#### 5.6.1.2 Terrorism Event Domain

To evaluate the multifaceted event recognition approach on the terrorism event domain, I used the MUC-4 data set [76], which is a standard benchmark collection for evaluating event extraction systems.

The documents in this corpus describe Latin American terrorist events including kidnapping, arson, bombing, and other attack events. Each document comes with associated answer key templates, a template per event. Specifically, I will consider a document as relevant if it has one or more associated answer key templates; otherwise, I will consider the document as irrelevant. Roughly half of the documents are relevant (i.e., they mention at least 1 terrorist event) and the rest are irrelevant.

The MUC-4 corpus consists of 1700 documents. When this data set was used for event extraction evaluations, researchers have split the data into training (DEV, 1300 documents), tuning (TST1+TST2, 200 documents), and test set (TST3+TST4, 200 documents). For my multifaceted event recognition evaluation, I will keep the same tuning set and test set. In addition, I will ignore annotations for the original training set and use the unannotated documents to learn event dictionaries.

---

[2] I used "marched" and "marching" as keywords but did not use "march" because it often refers to a month.

[3] These 400 documents were excluded from the unannotated data used for dictionary learning.

### 5.6.2 Metrics

The event recognition performance will be reported as Precision/Recall/F(1)-score. The Precision score is the number of correctly labeled event relevant documents divided by the total number of documents labeled by the event recognition system as event relevant. The Recall score is the number of correctly labeled event relevant documents divided by the total number of event relevant documents annotated in the data set. The F(1)-score is the harmonic mean of the Precision-score and the Recall-score.

### 5.6.3 Baselines

To demonstrate the effectiveness of the multifaceted event recognition approach, I will compare the recognition performance with two types of baselines. First, I designed two supervised learners. Both of them are classifier based (support vector machines (SVMs) [51] with a linear kernel [55]) and were trained using 10-fold cross validation with the test data set of each event domain. The first classifier used unigrams as features, while the second classifier used both unigrams and bigrams. All the features are binary.

Event recognition can be formulated as an information retrieval (IR) problem. As another point of comparison, I ran an existing IR system, Terrier [82], on the test set. Terrier was run with the parameter PL2 which refers to an advanced Divergence From Randomness weighting model [7]. In addition, Terrier used automatic query expansion.

## 5.7 Results on the Civil Unrest Event Domain

Because each article in the civil unrest test set contains at least one unrest event keyword, the first row of Table 5.4 shows the percentage of relevant documents in this data set, which reflects the accuracy of event keywords alone. Only 101 of the 300 documents in the test set were labeled as relevant by the annotators (i.e., 101 describe a civil unrest event). This means that using only the event keywords to identify civil unrest documents yields about 34% precision. In a second experiment, **KeywordTitle**, I required the event keyword to be in the title (headline) of the document. The KeywordTitle produced better precision (66%), but only 33% of the relevant documents had a keyword in the title.

The second section of Table 5.4 shows the results of the two supervised classifiers. We can see that the unigram classifier has an F-score of .64. Using both unigram and bigram features increased precision to 71% but recall fell by 7%, yielding a slightly lower F-score of .62.

**Table 5.4**: Experimental Results for Civil Unrest Event Recognition

| Method | Recall | Precision | F |
|---|---|---|---|
| *Keyword Accuracy* | | | |
| **Keyword** | - | 34 | - |
| **KeywordTitle** | 33 | 66 | 44 |
| *Supervised Learning* | | | |
| **Unigrams** | 62 | 66 | 64 |
| **Unigrams+Bigrams** | 55 | 71 | 62 |
| *Bootstrapped Dictionary Lookup* | | | |
| **Event Phrases (EV)** | 60 | 79 | 69 |
| **Agent Phrases (AG)** | 98 | 42 | 59 |
| **Purpose Phrases (PU)** | 59 | 67 | 63 |
| **Multifaceted** | 71 | **88** | **79** |

### 5.7.1 Event Recognition with Bootstrapped Dictionaries

Next, I used the bootstrapped dictionaries for event recognition. The bootstrapping process ran for 8 iterations and then stopped because no more phrases could be learned. The quality of bootstrapped data often degrades as bootstrapping progresses, so I used the tuning set to evaluate the quality of the dictionaries after each iteration. The best performance on the tuning set, based on the performance for the **Multifaceted** approach, resulted from the dictionaries produced after four iterations, so I used these dictionaries for the experiments.

Table 5.5 shows the number of event phrases, agents, and purpose phrases learned after each iteration. All three lexicons were significantly enriched after each iteration. The final bootstrapped dictionaries contain *623* event phrases, *569* purpose phrases, and *139* agent terms. By examining them manually, the learned phrases are highly diverse. Table 5.6 shows samples from each event dictionary. Appendix C gives more complete lists of the learned event phrases and facet phrases.

The third section of Table 5.4 shows the results when using the bootstrapped dictionaries for event recognition. I used a simple dictionary look-up approach that searched for dictionary entries in each document. I also explored various ways of using the bootstrapped dictionaries as features for a classifier to see if a supervised learner could make better use of

**Table 5.5**: Civil Unrest Dictionary Sizes after Bootstrapping

| | Event Phrases | Agent Terms | Purpose Phrases |
|---|---|---|---|
| **Iter #1** | 145 | 67 | 124 |
| **Iter #2** | 410 | 106 | 356 |
| **Iter #3** | 504 | 130 | 402 |
| **Iter #4** | 623 | 139 | 569 |

**Table 5.6**: Examples of Dictionary Entries for the Civil Unrest Event Domain

| |
|---|
| **Event Phrases:** went on strike, took to street, chanted slogans, gathered in capital, formed chain, clashed with police, staged rally, held protest, walked off job, burned flags, set fire, hit streets, marched in city, blocked roads, carried placards |
| **Agent Terms:** employees, miners, muslims, unions, protestors, journalists, refugees, prisoners, immigrants, inmates, pilots, farmers, followers, teachers, drivers |
| **Purpose Phrases:** accusing government, voice anger, press for wages, oppose plans, urging end, defying ban, show solidarity, mark anniversary, calling for right, condemning act, pressure government, mark death, push for hike, call attention, celebrating withdrawal |

the dictionaries. However, the classifiers' performance is inferior to the look-up approach. Although the phrases were learned based on syntactic analysis and only head words were retained for generality, I wanted to match dictionary entries without requiring syntactic analysis of new documents. So I used an approximate matching scheme that required each word to appear within 5 words of the previous word. For example, "held protest" would match "held a large protest" and "held a very large political protest". In this way, I avoid the need for syntactic analysis when using the dictionaries for event recognition.

First, I labeled a document as relevant if it contained any Event Phrase (EV) in the dictionary. The learned event phrases achieved better performance than all of the baselines, yielding an F-score of 69%. The best baseline was the unigram classifier, which was trained with supervised learning. The bootstrapped event phrase dictionary produced much higher precision (79% vs. 66%) with only slightly lower recall (60% vs. 62%), and did not require annotated texts for training. Statistical significance testing shows that the Event Phrase lookup approach works significantly better than the unigram classifier ($p < 0.05$, paired bootstrap [14]).

For the sake of completeness, I also evaluated the performance of dictionary look-up using the bootstrapped Agent (AG) and Purpose (PU) dictionaries, individually. The agents terms produced 42% precision with 98% recall, demonstrating that the learned agent list has extremely high coverage but (unsurprisingly) does not achieve high precision on its own. The purpose phrases achieved a better balance of recall and precision, producing an F-score of 63%, which is nearly the same as the supervised unigram classifier.

My original hypothesis was that a single type of event information is not sufficient to accurately identify event descriptions. My goal was high-accuracy event recognition by requiring that a document contain multiple clues pertaining to different facets of an

event (*multifaceted event recognition*). The last row of Table 5.4 (**Multifaceted**) shows the results when requiring matches from at least two different bootstrapped dictionaries. Specifically, I labeled a document as relevant if it contained at least one phrase from each of two different dictionaries and these phrases occurred in the same sentence. Table 5.4 shows that multifaceted event recognition achieves 88% precision with reasonably good recall of 71%, yielding an F-score of 79%. This multifaceted approach with simple dictionary look-up outperformed all of the baselines, and each dictionary used by itself. Statistical significance testing shows that the Multifaceted approach works significantly better than the unigram classifier ($p < 0.001$, paired bootstrap). The Multifaceted approach is significantly better than the Event Phrase (EV) lookup approach at the $p < 0.1$ level.

Table 5.7 takes a closer look at how each pair of dictionaries performed. The first row shows that requiring a document to have an event phrase and a purpose phrase produces the best precision (100%) but with low recall (14%). The second row reveals that requiring a document to have an event phrase and an agent term yields better recall (47%) and high precision (94%). The third row shows that requiring a document to have a purpose phrase and an agent term produces the best recall (50%) but with slightly lower precision (85%). Finally, the last row of Table 5.7 shows that taking the union of these results (i.e., any combination of dictionary pairs is sufficient) yields the best recall (71%) with high precision (88%), demonstrating that we get the best coverage by recognizing multiple combinations of event information.

We can see that the presented multifaceted event recognition approach adopts a fixed prescription to recognize event-relevant documents and it produces hard binary labels indicating if a document mentions a relevant event or not. However, depending on the application scenarios, we can easily vary the recognition method to attain different recall/precision tradeoffs or generate probabilities suggesting how likely a document is to describe a relevant event. For example, the earlier experimental results showed that we can achieve better accuracy by requiring matching of two pieces of event information. However, if we prefer high recall, we can simply tag documents that only contain an individual piece of event information. By loosening matching requirements, we would probably find more

**Table 5.7**: Analysis of Dictionary Combinations for Civil Unrest Event Recognition

| Method | Recall | Precision | F-score |
|---|---|---|---|
| **EV + PU** | 14 | 100 | 24 |
| **EV + AG** | 47 | 94 | 62 |
| **AG + PU** | 50 | 85 | 63 |
| **Multifaceted** | 71 | 88 | 79 |

documents that refer to a relevant event. For instance, by searching for event expressions or purpose phrases alone, we can find about 60 percent of relevant documents each with an acceptable precision.

Furthermore, instead of making hard decisions on the event relevance of documents, we could generate probabilities by weighing the contributions of each type of event information. For example, based on the results shown in Table 5.4, if a document has an event expression matched, then we could assign the probability .79 to it implying that there is a 79% chance it does describe a relevant event. If the same document has an additional event facet phrase matched, then we could assign higher probability to reflect the increased likehood that the document is event relevant. In addition, the presented multifaceted approach requires that two pieces of event information occur in a localized text region. What if we see two pieces of event information that are far apart in a document? This might lower our confidence that the document is event relevant; however, there is still a possibility that the document mentions a relevant event. In such cases, we can naturally use probabilities to signify our expectation that the document is event relevant, possibly based on the number of sentences between two pieces of event information in text.

### 5.7.2 Classifiers with Bootstrapped Dictionaries

I also explored the idea of using the bootstrapped dictionaries as features for a classifier to see if a supervised learner could make better use of the dictionaries. I created five SVM classifiers and performed 10-fold cross-validation on the test set.

Table 5.8 shows the results for the five classifiers. **TermLex** encodes a binary feature for every phrase in any of the dictionaries. **PairLex** encodes a binary feature for each pair of phrases from two different dictionaries and requires them to occur in the same sentence. The TermLex classifier achieves good performance (74% F-score), but is not as effective as the Multifaceted dictionary look-up approach (79% F-score). The PairLex classifier yields higher precision but very low recall, undoubtedly due to sparsity issues in matching specific pairs of phrases.

**Table 5.8**: Supervised Classifiers Using the Dictionaries

| Method | Recall | Precision | F-score |
|---|---|---|---|
| **TermLex** | 66 | 85 | 74 |
| **PairLex** | 10 | 91 | 18 |
| **TermSets** | 59 | 83 | 69 |
| **PairSets** | 68 | 84 | 75 |
| **AllSets** | 70 | 84 | 76 |

One of the strengths of my bootstrapping method is that it creates dictionaries from large volumes of unannotated documents. A limitation of supervised learning with lexical features is that the classifier can not benefit from terms in the bootstrapped dictionaries that do not appear in its training documents. To address this issue, I also tried encoding the dictionaries as set-based features. The **TermSets** classifier encodes three binary features, one for each dictionary. A feature gets a value of 1 if a document contains any word in the corresponding dictionary. The **PairSets** classifier also encodes three binary features, but each feature represents a different pair of dictionaries (EV+AG, EV+PU, or AG+PU). A feature gets a value of 1 if a document contains at least one term from each of the two dictionaries in the same sentence. The **AllSets** classifier encodes 7 set-based features: the previous six features and one additional feature that requires a sentence to contain at least one entry from all three dictionaries.

The **All Sets** classifier yields the best performance with an F-score of 76%. However, my straightforward dictionary look-up approach still performs better (79% F-score), and does not require annotated documents for training.

### 5.7.3 Comparisons with an Information Retrieval System

I used the Terrier information retrieval system to rank these 300 documents given my set of event keywords as the query. Specifically, I gave Terrier one query with all of the event keywords. Then, I generated a recall/precision curve (Figure 5.7) by computing the precisions at different levels of recall, ranging from 0 to 1 in increments of *.10*. We can see that Terrier identified the first 60 documents (20% recall) with 100% precision. However, precision dropped sharply after that. The circle in Figure 5.7 shows the performance of my bootstrapped dictionaries using the **Multifaceted** approach. At a comparable level of recall (71%), the multifaceted approach using the bootstrapped dictionaries yielded improvement of 34% in precision (88% vs. 54%).

### 5.7.4 Finding Articles with No Event Keyword

The learned event dictionaries have the potential to recognize event-relevant documents that do not contain any human-selected event keywords. This can happen in two ways. First, *378* of the *623* learned event phrases do not contain any of the original event keywords. Second, some event descriptions will contain a known agent and purpose phrase, but the event phrase will be unfamiliar.

I performed an additional set of experiments with documents in the Gigaword corpus that contain no human-selected civil unrest keywords. Following the multifaceted approach

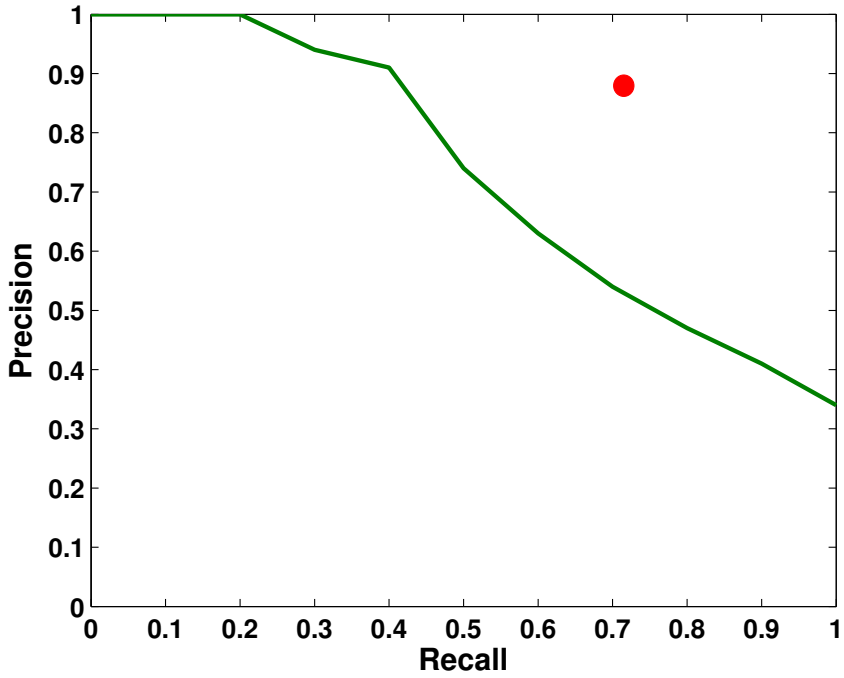


**Figure 5.7**: Comparison with the Terrier IR System, Civil Unrest Events

to event recognition, I collected all documents that contain a sentence that matches phrases in at least two of my bootstrapped event dictionaries. This process retrieved 178,197 documents. The Total column of Table 5.9 shows the number of documents that contained phrases found in two different dictionaries (EV+AG, EV+PU, AG+PU) or in all three dictionaries (EV+AG+PU).

I randomly sampled 50 documents from each category and had them annotated. The accuracies are shown in the Accuracy column. Finding all three types of phrases produced the best accuracy, 74%. Furthermore, I found over 6,800 documents that had all three types of event information using our learned dictionaries, but no event keywords. This result demonstrates that the bootstrapped dictionaries can recognize many event descriptions that would have been missed by searching only with manually selected keywords. This experiment also confirms that multifacted event recognition using all three learned

**Table 5.9**: Evaluation of Articles with No Event Keyword

| | **Total** | **Samples** | **Accuracy** |
|---|---|---|---|
| **EV+AG** | 67,796 | 50 | 44% |
| **EV+PU** | 2,375 | 50 | 54% |
| **AG+PU** | 101,173 | 50 | 18% |
| **EV+AG+PU** | 6,853 | 50 | **74%** |

dictionaries achieves good accuracy even for documents that do not contain the civil unrest keywords, although matching all facets is necessary to achieve good precision.

## 5.8 Results on the Terrorism Event Domain

I evaluated the bootstrapped terrorism event dictionaries on the test set (TST3 + TST4 sections) of the MUC-4 corpus. Out of the *200* documents, *126* articles mention one or more terrorism events. Therefore, if we label all the documents as event-relevant, the precision is only *63%* ( as shown in the first row of Table 5.10 ).

The second section of Table 5.10 shows the performance of the two supervised baselines. We can see that the unigram classifier yields a high recall of .86 and a reasonable precision of .76. Using both unigrams and bigrams as features, the supervised classifier further increases the recall to .91, but with a small loss of precision.

### 5.8.1 Event Recognition with Bootstrapped Dictionaries

To learn event dictionaries for the terrorism event domain, the bootstrapping process ran for 4 iterations and then stopped because no more phrases could be learned. Table 5.11 shows the number of event phrases, agents, patients, and effect phrases learned after each iteration. Nearly all event phrases and agents were learned from the first bootstrapping iteration while the number of patient and effect phrases gradually increased. The final

**Table 5.10**: Experimental Results for Terrorism Event Recognition

| **Method** | **Recall** | **Precision** | **F** |
|---|---|---|---|
| **Brute-force** | - | 63 | - |
| *Supervised Learning* | | | |
| **Unigrams** | 86 | 76 | **81** |
| **Unigrams+Bigrams** | 91 | 72 | 80 |
| *Bootstrapped Dictionary Lookup* | | | |
| **Event Phrases (EV)** | 19 | 65 | 29 |
| **Agent Phrases (AG)** | **99** | 63 | 77 |
| **Patient Phrases (PA)** | 94 | 64 | 76 |
| **Effect Phrases (EF)** | 94 | 68 | 79 |
| **EV + PA** | 17 | 75 | 27 |
| **EV + EF** | 17 | 75 | 27 |
| **EV + AG** | 17 | 70 | 27 |
| **PA + AG** | 90 | 65 | 76 |
| **EF + AG** | 87 | 69 | 77 |
| **PA + EF** | 85 | 71 | 78 |
| **EV + PA + EF** | 06 | **88** | 10 |
| **EV + AG + PA** | 14 | 77 | 23 |
| **EV + AG + EF** | 15 | 79 | 25 |
| **AG + PA + EF** | 46 | 85 | 60 |
| **Multifaceted (all Triples)** | 53 | **82** | 65 |

**Table 5.11**: Terrorism Dictionary Sizes after Bootstrapping

| | Event Phrases | Agent Terms | Patient Terms | Effect Phrases |
|---|---|---|---|---|
| **Iter #1** | 123 | 25 | 10 | 30 |
| **Iter #2** | 123 | 25 | 28 | 38 |
| **Iter #3** | 124 | 25 | 30 | 44 |
| **Iter #4** | 124 | 25 | 30 | 46 |

bootstrapped dictionaries contain *124* event phrases, *25* agent terms, *30* patient terms, and *46* effect phrases. Similar to the previous civil unrest event domain, phrases in the learned terrorism event dictionaries are highly diverse too. Table 5.12 shows samples from each event dictionary. Appendix D gives more complete lists of the learned event phrases and facet phrases.

The third section of Table 5.10 shows the experimental results when using the bootstrapped dictionaries for event recognition. As with the experiments for the civil unrest event domain, I used the simple dictionary look-up approach that searched for dictionary entries in each document. Furthermore, I used the same approximate matching scheme as discussed in Section 5.7.1.

The first four rows of the bottom section of Table 5.10 show the results of event recognition where a document is labeled as relevant if it contains at least one event phrase (the first row) or at least one facet phrase (the following three rows). We can see that requiring matching with only one type of event information gives mediocre precision. Interestingly, while matching with each type of facet phrase consistently yields high recall, matching only with event phrases recognizes only 19% of the relevant documents. This can be partially

**Table 5.12**: Examples of Dictionary Entries for the Terrorism Event Domain

| |
|---|
| **Event Phrases:** claimed responsibility, hit houses, burned trucks, blew up bus, set off bomb, holding hostages, killed citizens, threatened investigators, entered residence sabotaged tower, machinegunned residence, detonated charge, carry out attacks, attacked studios, massacred women |
| **Agent Terms:** mob, ESA, group, individual, commandos, Tegucigalpa, Santiago, organizations, groups, organization, forces, Willka |
| **Patient Terms:** headquarters, citizens, officer, leaders, neighborhood, reporters, airplanes, population, Ellacuria, leader, home, buildings, office |
| **Effect Phrases:** wounded *, * be destroyed, death of *, broke into *, killing of *, bodies of *, * be kidnapped, * be assassinated, enter *, set fire to *, * be wounded, causing damage to *, * be shot, massacre of *, * be detained |

attributed to the strict syntactic forms required for event phrases, but it also reflects the fact that terrorism event descriptions are of high diversity and using event phrases alone can only identify a small fraction of relevant documents.

The next six rows of the third section show the performance of event recognition where a document is labeled as relevant if it contains at least one phrase from each of two distinct event dictionaries. In addition, consistent with the assumption used in the learning process that multiple pieces of event information should co-occur in a localized text region, the matched phrases should appear in a text segment that spans at most four sentences. From the first three combinations, we can see that on top of event phrases, each type of facet phrases clearly improves the event recognition precision. Specifically, with a small recall loss, patient phrases and effect phrases each increase the recognition precision by 10 percent, and agent phrases improve the recognition precision by 5 percent. The latter three combinations show that when paired with another event facet, each event facet becomes more precise in recognizing the relevant event.

However, in general, the precisions obtained by matching two types of event information is still not satisfying. The reason is that we have identified four types of event information, event phrases, and three types of event facet phrases, that are all important to define terrorism events, and seeing only two types of information in a document is not so sufficient yet for us to claim that the document is to describe a relevant event. The following four rows show the recognition performance where a document is labeled as relevant if it contains at least one event phrase together with two event facet phrases of distinct types, or three facet phrases of distinct types. Furthermore, each three types of event information should co-occur in a text segment that spans at most four sentences. Compared to searching for only one or two types of event information in a document, each combination here further improves the event recognition precision.

The last row **Multifaceted (all Triples)** shows the results when matching all combinations of three types of event information. We can see that this approach achieves the best precision .82, which is higher than matching with only one or two types of event information and higher than both supervised baselines. One interesting observation is that out of the four combinations, the first and fourth combinations, which used both patient and effect dictionaries, achieve the best precisions. This confirms that characteristic patients together with the effects of patients provide useful indicators when identifying terrorism events.

Overall, on the terrorism domain, the performance of the multifaceted event recognition approach does not compare favorably with the supervised learning baselines. While the

precision of the multifaceted event recognition method is good, the recall is only 53% and around half of the relevant documents are missed. The F-score is around .15 lower than the two supervised baselines. One reason for the low recall is that the event dictionary learning process only used a limited amount of unannotated documents, 1300 documents (MUC-4 DEV) specifically, which is a much smaller collection compared to the Gigaword corpus used for the civil unrest event dictionary learning. The main obstacle of using the broad coverage Gigaword corpus for the MUC-4 terrorism domain dictionary learning is that the MUC-4 data set used for evaluation mostly describes terrorism events that happened in specific geographic areas (i.e., Latin American countries) and in a specific time period (largely 1990s). Therefore, the domain is full of special vocabulary referring to specific locations or terrorism organizations associated with Latin America that are not well-represented in the Gigaword corpus.

### 5.8.2 Comparisons with an Information Retrieval System

I used the Terrier information retrieval system to rank the 200 documents in the test set, given a set of event keywords as the query [4], and then generated a recall/precision curve (Figure 5.8) by computing the precisions at different levels of recall, ranging from 0 to 1 in increments of *.10*. With the given query, Terrier can only retrieve about 80 percent of the relevant documents. The circle in Figure 5.8 shows the performance of my bootstrapped dictionaries using the **Multifaceted** approach. We can see that at a comparable level of recall (55%), the multifaceted approach using my bootstrapped dictionaries yielded about a 10 percent improvement in precision (80% v.s. 71%).

## 5.9 The Effects of Multifaceted Event Recognition on Event Extraction

One strong motivation of studying event recognition is to further improve event extraction accuracy by focusing attention of extraction systems on documents that are deemed to contain domain relevant events. In this section, I will evaluate how my multifaceted event recognition approach impacts event extraction performance.

There can be different ways for incorporating the output of event recognition systems to influence extraction decisions. For example, we can apply event recognition as a hard filter-

[4]The event keywords are chosen based on the terrorism event subtypes as annotated in MUC-4 answer keys, and I included keywords for all four subtypes and their syntactic variations. They are: attacked, attack, attacks, bombed, bomb, bombs, bombing, kidnapped, kidnap, kidnapping, and arson. I gave Terrier one query with all of the event keywords.

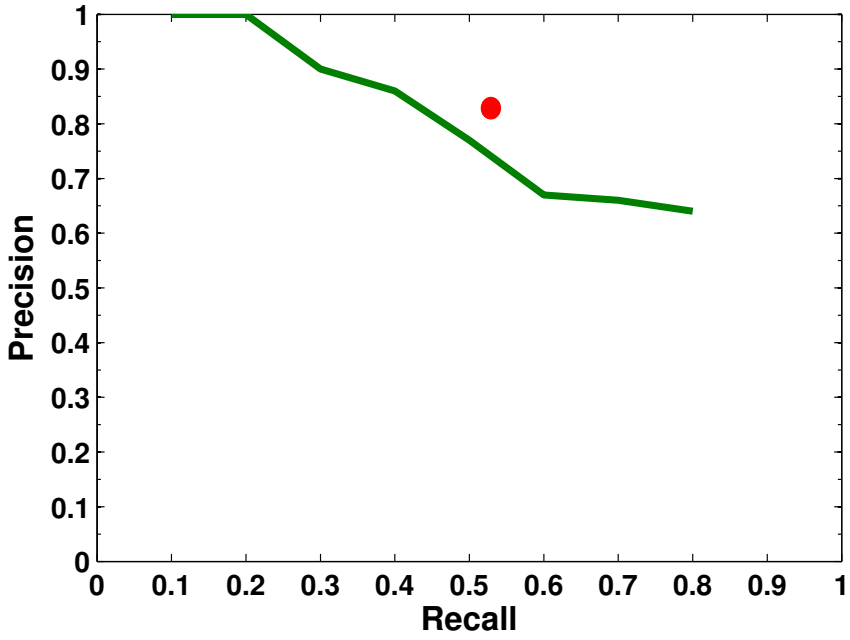


**Figure 5.8**: Comparison with the Terrier IR System, Terrorism Events

ing step and extraction systems afterwards on the filtered documents only. Alternatively, we can combine recognition and extraction results probabilistically to form the final extraction decisions if both phases generate probabilities. Because my multifaceted event recognition generates hard labels indicating if a document describes a relevant event, I will show event extraction results when applying the multifaceted event recognition as a document filter on top of my best discourse-guided event extraction architecture *LINKER*. Please refer to Chapter 4 for details on the design of *LINKER*. The expectation is that the precision of event extraction will be improved after the multifaceted event recognition approach filters out documents that do not mention a relevant event.

Tables 5.13 and 5.14 show the extraction results for the civil unrest event domain and the terrorism event domain, respectively. The first row of Table 5.13 shows the results of *LINKER* when it was applied to every document of the civil unrest test set, showing the extraction results for each event role separately and their macro average performance. These results are the same as the results of the full *LINKER* system in Table 4.2. Please refer to the evaluation sections of Chapter 4 for details on experimental settings. The second row of Table 5.13 shows the results of *LINKER* when it was only applied to the relevant documents as identified by my multifaceted event recognition system. We can see that the extraction precisions are improved across all four event roles. On average, the precision increases by 9

**Table 5.13**: Experimental Results on the Civil Unrest Domain, Precision/Recall/F-score

| System | Agent | Site | Location | Instrument | Average |
|---|---|---|---|---|---|
| **LINKER** | 58/41/48 | 43/27/33 | 41/33/36 | 64/61/63 | 51/40/45 |
| **+Multifaceted** | 65/34/45 | 54/23/32 | 55/29/37 | 68/61/65 | 60/37/46 |
| with Perfect Document Classifier | | | | | |
| **+PerfectDoc** | 65/41/50 | 54/27/36 | 54/33/41 | 69/61/65 | 61/40/48 |

**Table 5.14**: Experimental Results on the Terrorism Domain, Precision/Recall/F-score.

| System | PerpInd | PerpOrg | Target | Victim | Weapon | Average |
|---|---|---|---|---|---|---|
| **LINKER** | 54/57/56 | 55/49/51 | 55/68/61 | 63/59/61 | 62/64/63 | 58/60/59 |
| **+Multifaceted** | 62/42/50 | 56/42/48 | 51/42/46 | 63/45/52 | 56/36/43 | 59/42/49 |
| with Perfect Document Classifier | | | | | | |
| **+PerfectDoc** | 64/57/61 | 57/49/53 | 60/68/64 | 68/59/63 | 65/64/64 | 64/60/62 |

points with only 3 points of recall loss. Due to the substantial precision improvement, the F-score is also slightly increased.

After the document level filtering using the multifaceted event recognition approach, the precision of the event extraction system *LINKER* is still relatively low (only 60%). Therefore, to see the maximum of extraction precision gain by applying event recognition on top, I showed the extraction results (in the last row of Table 5.13) when a perfect document classifier is applied on top of the extraction system. By the perfect document classifier, I will simply apply *LINKER* only to the gold standard relevant documents in the test set. We can see that, with perfect event recognition, the extraction recall will be the same as applying the extraction system to each document in the test set. The precision was improved by only one further point, compared to the setting where my multifaceted approach was employed.

Table 5.14 shows the event extraction results for the terrorism domain. The first row shows the results of *LINKER* when it was applied to every document of the terrorism test set. These results are the same as shown in Table 4.1. Please refer to Section 4.4 for details on experimental settings. The second row of Table 5.14 shows the extraction results of *LINKER* when it was only applied to the relevant documents identified by my multifaceted event recognition approach. For the terrorism domain, we see only a slight improvement of precision after applying the document level filter. However, the extraction recall was substantially reduced. Similar to the last previous table, the last row of Table 5.14 also shows *LINKER*'s extraction results when the perfect document classifier is applied on top of the extraction system. We can see that with perfect event recognition, the extraction precision increased to just .64 for the terrorism domain.

Another benefit of applying event recognition before actually diving into documents

for event extraction is that amount of computing resources will be saved. Specifically, for the civil unrest event domain, only *83* documents were processed by the event extraction system, compared to *300* documents in the original test set. Similarly for the terrorism event domain, only *85* documents needs to be processed by the event extraction system, instead of *200* documents as included in its full test set. Considering the costly text analysis components that are used in event extraction systems, event recognition will increase the throughtput rate of event extraction systems in practice.

## 5.10 Conclusions

In this chapter, I presented my multifaceted event recognition approach that can accurately recognize event descriptions of a particular type in text by identifying event expressions as well as event defining characteristics of the event. I also presented a bootstrapping framework to automatically learn event expressions as well as essential facets of events, which only requires limited human supervision. Experimental results show that the multifaceted event recognition approach can effectively identify documents that describe a certain type of event and make event extraction systems more precise.

# CHAPTER 6

# CONCLUSIONS AND FUTURE DIRECTIONS

In this chapter, I will first summarize my event extraction and recognition research and contributions. Then, I will discuss several future directions that may lead to futher improved event extraction and recognition performance.

## 6.1 Research Summary and Contributions

My research has been concentrated on improving event extraction performance by explicitly identifying event contexts before extracting individual facts. In this section, I will first emphasize the problems and limitations of current event extraction systems, which have motivated my research, then I will briefly go through the approaches and algorithms that I introduced to improve event extraction performance and highlight my contributions.

Most current event extraction systems heavily rely on local contexts and individual event expressions to recognize event mentions when making extraction decisions. However, lacking the view of wider context limits both the extraction coverage and the accuracy of traditional event extraction systems. The coverage of event extraction systems is limited because many role fillers occur in contexts that do not explicitly mention the event, e.g., the perpetrator of a murder may be mentioned in the context of an arrest, and those fillers are often overlooked. The accuracy of current event extraction systems is limited too because even if the local context contains seemingly relevant event keywords or phrases. depending on the larger context, they may not be referring to a relevant event due to ambiguity and metaphor. For example, "Obama was attacked" may lead to Obama being extracted as the victim of a physical attack, even if the preceding sentences describe a presidential debate and the verb "attacked" is being used metaphorically. Therefore, by only considering local contexts, current event extraction systems can generate many false extractions.

To address these limitations of current event extraction systems and improve event extraction performance, I proposed two new event extraction architectures that incorporate discourse information across sentences to recognize event contexts before applying local

extraction models. First, to seek event information out of secondary contexts and thus improve the coverage of event extraction performance, I created *TIER*, a multilayered event extraction architecture that performs document level, sentence level, and noun phrase level text analysis to progressively "zoom in" on relevant event information. The challenge for extracting information from secondary event contexts is that secondary contexts occur with irrelevant events too. For example, an arrest can follow a theft instead of a terrorism event. Keeping this in mind, *TIER* represents a two-pronged strategy for event extraction that distinguish two types of documents that mention relevant events, *event narratives* vs. *fleeting references*, and only extracts information from secondary contexts in *event narratives*. Event narratives are articles whose main purpose is to report the details of an event while fleeting references are the documents that only briefly mention a relevant event, but do not elaborate on the event details.

To make event extraction systems more precise, I also proposed a discourse-guided event extraction model, called *LINKER*. In addition to a set of local role filler extractors as normally seen in event extraction systems, *LINKER* includes a structured sentence classifier that sequentially reads a story and determines which sentences contain event information based on both the local and preceding contexts. Then, the structured sentence classifier and the set of local role filler extractors are combined by extracting only the candidate role fillers that occur in sentences that represent event contexts, as determined by the sentence classifier. Specifically, the structured learning algorithm, conditional random fields (CRFs), explicitly models whether the previous sentence is an event context, which captures discourse continuity across sentences. Furthermore, the structured sentence classifier uses well-designed features to capture textual cohesion properties across sentences, including lexical word associations, discourse relations across sentences, and distributional properties of the candidate role fillers within and across sentences.

Another issue of current event extraction systems is that they do not attempt to determine whether a document contains any relevant information before extracting facts based only on local context. Processing documents that do not mention a relevant event is a waste of computing resources. Furthermore, accurate event recognition will improve event extraction accuracy because any extractions from irrelevant documents will be false. However, identifying documents that mention an event of a particular type is a highly challenging task due to the high complexity and variety of event descriptions. Event keywords are rarely reliable on their own. For example, consider the challenge of finding documents about civil unrest. The words *"strike"* and *"rally"* refer to common types of

civil unrest, but they frequently refer to other things as well. A strike can refer to a military event or a sporting event (e.g., *"air strike"*, *"bowling strike"*), and a rally can be a race or a spirited exchange (e.g., *"car rally"*, *"tennis rally"*). I proposed multifaceted event recognition to accurately recognize event descriptions in text by identifying event expressions as well as event facets, which are defining characteristics of the event. Event facets, such as agents, purpose, and effects of events, are essential to distinguish one type of event from another. For example, given the event expression "hit the village", depending on the agents, it might refer to a natural disaster event if the agent is "The flooding", or it might be describing an air strike if the agent is "The military bombs".

I also proposed a bootstrapping framework to automatically learn event expressions as well as essential facets of events, which relies on limited human supervision. The learning algorithm exploits the observation that event expressions and event facet information often appear together in sentences that introduce an event. Furthermore, seeing more than one piece of event information in a sentence tends to validate that the sentence is an event sentence and suggests that additional event information may also be in the same sentence. Therefore, the bootstrapping algorithm ricochets back and forth, alternately learning new event phrases and learning new event facet phrases, in an iterative process.

After reflection on my research, I have focused on improving coverage and accuracy of event extraction systems by recognizing event contexts before applying extraction models. I investigated event context identification by both recognizing documents that mention a relevant event and finding event regions within a document. Specifically, my main contributions are as follows:

1 *I distinguish secondary event contexts from primary contexts and propose a multilayered event extraction architecture that can seek out event information from different types of event contexts.*

   Many event role fillers occur in secondary contexts that do not explicitly mention the event and are generally not part of the main event description ("primary contexts"). Event role fillers in secondary contexts are generally overlooked by current event extraction systems. To seek event information out of secondary contexts, I introduced *TIER*, a multilayered event extraction architecture that performs document level, sentence level, and noun phrase level text analysis to progressively "zoom in" on relevant event information.

2 *I proposed a discourse-guided event extraction architecture that uses a single structured event sentence classifier to capture various textual cohesion properties and identify*

*event contexts in a document.*

The discourse-guided event extraction architecture is called *LINKER*. In addition to a set of local role filler extractors, *LINKER* uses a single sequentially structured sentence classifier to explicitly model the contextual influences across sentences and identify event-related story contexts. In the structured sentence classifier, a variety of discourse information is modeled as textual cohesion properties across sentences, including lexical cohesion features, discourse relations, and domain-specific candidate role filler distributional features.

3 *I proposed multifaceted event recognition, which uses event defining characteristics, in addition to event expressions, to identify documents describing a particular type of event.*

Finding documents that describe a particular type of event is a challenging task due to the high complexity and variety of event descriptions. Event keywords tend to be ambiguous and are not sufficient to identify documents that discuss events of a specific type. I observed that event defining characteristics, I call them event facets, such as agents, purpose, and effects of events, are essential to distinguish one type of event from another. Therefore, I proposed multifaceted event recognition to accurately recognize event descriptions in text by identifying event expressions as well as event facets. I also proposed a bootstrapping framework to automatically learn event expressions as well as essential facets of events from free texts, which only requires minimal human supervision.

## 6.2 Future Directions

Evaluation of my research using two distinct event domains has shown that the discourse-guided event extraction architectures can improve both coverage and accuracy of event extraction performance and the multifaceted event recognition can effectively identify event-relevant documents and further improve event extraction accuracy. However, we can see that overall, the extraction performance is far from perfect. Furthermore, most event extraction systems heavily rely on human annotated data to train event extraction systems for each type of event. It is important to reduce the dependence on human supervision to be able to quickly configure domain-specific event extraction systems. In the following subsections, I will discuss several thoughts I have when I meditate on the research presented in this dissertation and present some ideas that may lead to better event extraction performance or reduce human supervision that is required to train event extraction systems.

### 6.2.1 Incorporating Domain Knowledge to Tackle Inferences

Most current event extraction systems are automatically trained using human-labeled data as supervision and use limited amounts of external domain knowledge. Therefore, systems trained this way heavily rely on surface textual clues, word forms, or shallow semantics of words, to determine if an event is being described and what type of event information is being conveyed. For instance, if a person has been annotated as a victim of terrorism events several times and each time, the textual contexts that follow the extractions are the same, e.g., "was killed", then the automatically trained event extraction systems can learn the pattern that whenever a person was followed by "was killed", the person should be extracted as a victim. However, this is clearly not sufficient to imitate how humans process information.

Generally, we go through various kinds of inferences at multiple levels when we read texts and digest information described in texts. For example, suppose a document describes a bombing event in a shopping mall and later discusses a police investigation on a man in a hat that appeared in the monitored video of the shopping mall before the bombing event happened. We can easily infer that the investigated man has been suspected to have carried out the bombing and should be extracted as the potential perpetrator of the bombing event. In my work, I distinguish secondary contexts from primary contexts and proposed the multilayered event extraction system, *TIER*, to seek out information from the contexts that do not explicitly refer to the event. In this case, the investigation context can be viewed as a secondary context. However, without plugging in explicit inference driven components, the automatically trained event extraction systems are good at recognizing *textual patterns* that have repeated themselves many times, but tend to miss the less frequent ones. Unfortunately, due to the diversity of event descriptions, a large proportion of textual clues only occur occasionally in the annotated data.

However, with the aid of a rich set of domain knowledge, we can design extraction components and mechanisms that carry out inferences similar to the human. Therefore, we possibly obtain opportunities to automatically make intelligent extraction decisions with many less common contexts. In the next section, I will present my thoughts on the specific types of domain knowledge that may be helpful to event extraction.

### 6.2.2 Acquiring Domain Knowledge from Unlabeled Texts

As discussed in the previous subsection, inference driven event extraction might enable better extraction performance. Ideally, we should design algorithms that can acquire domain knowledge from a large volume of unlabeled and easily accessible plain texts, which can

reveal *textual patterns* that may have appeared only one time in a limited size of annotated data set.

The primary question is what type of domain knowledge we should aim to acquire. The first category of domain knowledge I have identified is events that are somehow related to the target event. Events are often causally linked, temporally connected, or mutually influenced. This nature of event is also manifested in their textual descriptions. For instance, articles that mainly describe an event also discuss its related events that may have caused the target event or happened as the consequences of the target event. Therefore, going beyond examining the descriptive nature of individual events, studying the relations between different types of events will help us to better detect the relevant event descriptions and locate event information. I am especially interested to answer the question of how one event occurs in the contexts of a variety of the other events and what are the types of associations among events.

The second type of domain knowledge that may benefit event extraction performance is subevents of the target event. I have observed that in event descriptions, event information is frequently mentioned in detailed elaborations of how the event happened. Therefore, knowledge about routine subevents of a particular type of event should be helpful for event extraction systems to identify specific types of event information.

### 6.2.3 Building Event Ontologies

Most current event extraction systems are trained for predefined types of events. Mainly, training event extraction systems to tackle certain type of events can effectively reduce the complexity of text analysis needed to detect and extract event information; however, it also implies that the extraction systems should be retrained whenever new types of events are targeted. To resolve this dilemma, one possible solution is to build up an event ontology and associate the automatically acquired domain knowledge to its corresponding type of event in the event ontology.

To achieve this goal, we have to answer questions about the structure of an event ontology. For example, what criteria should we use to categorize events and how many main event classes should be included in the event ontology. Possibly, we can use event facets to group events and events sharing the same set of essential event facets form a class of events. The other type of questions are about when we should build links between event classes and what the interevent relations can be. Potentially, we can use the same set of interevent relations as discussed in Subsection 6.2.2, including causal, temporal relations and the relation that an event is a subevent of the other event.

In short, a well-structured event ontology and a rich collection of domain knowledge that is mapped to each type of event in the ontology are valuable to both further improve the performance of event extraction systems and reduce their dependence on human supervision. First, domain knowledge of a specific type of event and domain knowledge of its related events can be explored to develop inference driven event extraction systems. And these systems have potential to make intelligent extraction decisions and enhance extraction performance. Second, event domain knowledge and structured ontology represent generalizations and summaries of diverse forms of events; therefore, the access of such knowledge can make the system training less dependent on human supervision that is often realized by annotating specific event descriptions.

# APPENDIX A

# CU EVENT DOCUMENT ANNOTATION GUIDELINES

This appendix lists the civil unrest event document annotation guidelines that were provided to the annotators.

## A.1 Annotation Task Definition

You will need to read a set of news articles and determine which articles discuss a CU event. If an article mentions at least one CU event, label it as a CU_article. If an article mentions no CU event, label it as an Irrel_article.

## A.2 Civil Unrest Event Definition

Civil unrest (CU) is a broad term that is typically used by the media or law enforcement to describe a form of public disturbance caused by a group of people for a purpose. Civil unrest events include activities to protest against major socio-political problems, events of activism to support a cause (e.g., peace rallies or large-scale marches to support a prominent figure), and events to promote changes in government or business affairs (e.g., large gatherings to rally for higher wages). Types of civil unrest can include, but are not necessarily limited to: strikes, rallies, sit-ins and other forms of obstructions, riots, sabotage, and other forms of public disturbance motivated by a cause. It is intended to be a demonstration to the public, the government, or an institution (e.g., business or educational sectors), but can sometimes escalate into general chaos.

### A.2.1 Civil Unrest Events to Be Annotated

**1.** According to the definition, CU events do not include war, ethnic fightings, or fightings involving armed parties only.

**2.** CU events include mentions of currently on-going and recent (within one year) CU events. Old CU events that happened more than one year ago from the date when the article was published should not be labeled.

3. CU events include mentions of threatened and planned CU events that may happen even if there is uncertainty or they are mentioned conditionally. CU events do not include mentions of threatened and planned CU events that will definitely not happen.

4. CU events do not include purely hypothetical or abstract mentions of CU events or activities. These events are mentioned in general discussions or metaphorically.

5. CU events can be described only in a small portion of a text and may not be the focus of the text. However, if an article only mentions CU events in a single noun phrase fleetingly, e.g., "last year's teachers strike", "a possible student protest", no any other detail about those events are mentioned, they are treated as abstract mentions of CU events and the article is an Irerel_article.

6. Event summary information that is synthesized from two or more events should NOT be annotated.

# APPENDIX B

# CU EVENT ROLE FILLER ANNOTATION GUIDELINES

This appendix lists the civil unrest event role filler annotation guidelines that were provided to the annotators.

## B.1 Annotation Task Definition

You will need to read a set of news articles and identify the CU event descriptions. Specifically, you need to find out phrases in text that describe CU events and put them into their corresponding slots. Each slot indicates one type of event information, such as locations, agents, causes, and damages of CU events.

### B.1.1 Notes

1. More than one event can be described in an article; however, only one set of event role slots are to be filled out. Therefore, you should put all the phrases fulfilling a specific event role into the same slot EVEN IF THEY ARE FROM DIFFERENT EVENTS.

2. Events that are described in an article are often related. If the same entity or object is involved and plays the same role in multiple events, you should only annotate the mentions across all its mentions that are significantly different in lexical forms.

## B.2 The Event Slots to Be Annotated

Six event slots and event subtypes will be considered for annotations.

**(1) Event Type (Closed Set)**

Instead of labeling strings in text like (1), choose event types from the following CU types:

STRIKE(S) – consists of refusing to work.

MARCH(ES) – consists of moving from one place to another place.

SIT-IN(S)/OCCUPATION(S) – consists of taking up some space and thus disturbing

the regular activities that require the space.

OTHER(S) – other forms of protest(s) or demonstration(s), such as rally (rallies), riot(s), sabotage(s), etc.

If one CU event includes activities of type STRIKE (or MARCH, or OCCUPATION), select STRIKE (or MARCH, or OCCUPATION) as the event type, even if the event described also includes activities of other types. If one event includes multiple types of activities, e.g., both STRIKE and OCCUPATION, select all the appropriate types. Select types for all CU events described in an article, e.g., select both STRIKE and MARCH if two CU events were described in an article, if one event is of type STRIKE and another is of type MARCH. You should select OTHER if none of the CU events described in an article includes activities of any of the first three types.

**(2) Agents**

The population GROUPS who initiate, lead, or join to strengthen the CU events. If there are multiples references to roughly the same population group, label all the ones that are in different lexical forms, including the general mention terms such as "the protesters".

**(3) Sites**

A human constructed facility where a CU event takes place. A site can be a plaza, a shopping mall, a mosque, a bridge, a hospital, or an university.

**(4) Locations**

NAMED geographical regions/areas where a CU event takes place. A location can be a city (e.g., "Beijing"), a country (e.g., U.S.) or other named places (e.g., Antelope Island). Only label the location names themselves. You should only consider the locations that appear in the context of a CU event. You should not consider the locations that are embedded in organization names. For instance, in

*25,000 opposition supporters demonstrated in Lome, the capital of Togo.*

you should label "Lome" and "Togo" as two locations and put them in two lines.

**(5) Victims**

The casualties that are *due to the CU events* can refer to any people that are injured or died in the civil unrest events, including *both agents and other types of people.* If you are not certain based on the text descriptions that some people were injured or died, do not annotate them. For example, if you see

*Two guards were hit by stones.*

in the context of the CU events, "Two guards" should be annotated because people generally get hurt when hit by stones. However, given

*A policeman was hit by eggs.*

in the context of the CU events, "A policeman" should not be annotated because people generally would not be hurt when hit by eggs.

**(6) (Affected) Facilities**

Buildings or property that are physically damaged or destroyed *due to the CU events*. For instance, given

*They stormed an airport, damaged the VIP lounge and surged onto the runway to prevent a flight taking off.*

in the context of the CU events, you should label "the VIP lounge" as one filler.

**(7) Instruments**

Anything that is used by *both agents and other types of people during the CU events* with the intent to injure others, damage property, control the crowd, or defend themselves. Weapons can be stones, bombs batons, and tear gas. For instance, if you see

*Police used tear gas, fire hoses and pepper spray to hold back hundreds of demonstrators led by militant Korean farmers, some of whom were armed with bamboo sticks and metal bars .*

in the context of the CU events, you should label "tear gas", "fire hoses", "pepper spray", "bamboo sticks", and "metal bars" as the weapons.

### B.2.1 Notes

1. Please label appropriate strings in both headlines and body texts, but with preference to strings from body texts. If string A from the headline and string B from the body text are equally informative to be a role filler, please label string B instead of A. If an appropriate role filler C is only seen in the headline, please label C.

2. For slots Agent/Population, Site/Facility, HumanEffects, PhysicalEffects and Instruments/Weapons, annotations should be complete base noun phrases that include head nouns, modifiers, determiners, and articles.

# APPENDIX C

# BOOTSTRAPPED EVENT PHRASES AND EVENT FACET PHRASES FOR THE CU DOMAIN

This appendix lists more samples from bootstrapped event dictionaries for the civil unrest event domain, including event phrases and phrases for two event facets: agents and purpose.

**Table C.1**: Bootstrapped Agent Terms for the CU Domain

| |
|---|
| **Agent Terms:** employees, miners, muslims, unions, protestors, journalists, refugees, prisoners, immigrants, inmates, pilots, farmers, followers, teachers, drivers, professors, villagers, cypriots, fundamentalists, tamils, syrians, marchers, nurses, residents, argentines, radicals, kurds, tribesmen, shiites, youths, expatriates, iraqis, iranians, unionists, exiles, albanians, maoists, retirees, tibetans, venezuelans, settlers, haitians, uighurs, migrants, lawyers, veterans, hard-liners, pensioners, servants, growers, reservists, fishermen, afghans, defendants, truckers, campaigners, communists, clerics, italians, extremists, leftists, relatives, loyalists, rioters, koreans, romanians, colombians, serbs, monks, hindus, organizations, intellectuals, guards, laborers, dissidents, gypsies, peasants, turks, macedonians, indians, police, firefighters, strikers, hundreds, arabs, mourners, autoworkers, traders, staff, soldiers, israelis, worshippers, priests, members, thousand, chileans, metalworkers, christians, factions, practitioners, survivors, officers, judges, dockworkers, kashmiris, pilots, islamists, moslems, hardliners, prostitutes, thousands, citizens, graduates, pakistanis, owners, kongers, environmentalists, |

**Table C.2**: Bootstrapped Purpose Phrases for the CU Domain

| |
|---|
| **Purpose Phrases:** urging end, voice anger, press for wages, oppose plans, condemning act, show solidarity, mark anniversary, calling for right, mark death, accusing government, pressure government, push for hike, celebrating withdrawal, call attention, voice opposition, prevent closure, press demand, express anger, show dissatisfaction, commemorate anniversary, venting anger, press for release, calling dictator, denounce crackdown, contest policies, demonstrate opposition, protect jobs, denouncing israel, show opposition, forcing police, show support, underscore demands, oppose decision, show support, express opposition, mourn deaths, press for resignation, prompting police, calling for government, drawing thousands, denounce government, demonstrate against government, resume imports, opposing war, forcing cancellation, press for pay, hurling insults, calling for resignation, defying curfew, denounce decision, call for end, force government, commemorate death, flouting refusals, vent anger, oust chavez, disrupting traffic, press for increase, press for reforms, denounce measures, condemning bush, resume testing, press for rise, chanting shame, call for ouster, expand settlements, denounce attacks, oppose plans, demonstrate support, stopping services, denouncing violence, denounce killings, paralysing country, blaming britain, call for action, denounce law, denouncing rule, calling for peace, opposing visit, highlight plight, denouncing states, voice protest, disrupting services, support demands, denounce victory, raise salaries, press pay, paralyzing production, stranding passengers, prevent clashes, paralyzing activities, blaming death, mark start, press for increases, halting flights, press for conditions, seek wages, vent fury, calling for scrapping, paralyzing traffic, press for hikes, demonstrating against detention, push for pay, press for payment, disrupting production, causing chaos, turn up pressure, causing disruptions, push demands, back demands, forcing authorities, accusing authorities, celebrate attack, condemn handling, pay cuts, privatize network, causing shortages, interrupting services, paralyzing services, press management, urge members, shutting down traffic, denounce conditions, idling buses, shouting justice , press for freedom, express rage, press elections, call for reforms calling minister, express outrage, chanting support, press claim, press for negotiations, heightening pressure, launching challenge, paralysing city, prompting guards, denounce war, keeping up pressure, threatening disruption, defend allowance, shouting with government, criticize government, reject terrorism, preventing lawmakers, call for reunification |

**Table C.3**: Bootstrapped Event Phrases for the CU Domain

| |
|---|
| **Event Phrases:** went on strike, took to street, chanted slogans, set fire, formed chain, clashed with police, staged rally, held protest, walked off job, burned flags, hit streets, marched in city, blocked roads, carried placards, marched through paris, occupied office, rallied outside house, carried signs, gathered in capital, staged strike, held demonstration, protested outside office, waved flags, unfurled banner, marched kilometers, demonstrated in city, defy ban, went on rampage, blocked streets, held vigil, carried banners, declared strike, marched through center, were on strike, launched strikes, held demonstrations, called for boycott, waved banners, stage strike, begun strike, staged sit-in, staged series, rallied near embassy, rallied in capital, demonstrated in capital, plan strike, gathered in center, began protest, blocked highways, occupied embassy, rallied in city, are on strike, protested in capital, rallied outside embassy, marched to office, filled streets, stepped up protests, marched in cities, blocked access, marched in capital, staged rallies, go on strike, burned tires, occupied headquarters, poured into streets, burned effigies, called for strikes, smashed windows, gone on strike, laid siege, marched through city, gathered in square, marched to station, picketed headquarters, observed strike, blocked traffic, blocked entrance, staged walkouts, held banners, demonstrated outside parliament, blockaded port, stopped work, joined protests, attended rallies, organized rally, rallied outside ministry, marched on palace, gathered outside church, flooded streets, continued protest, ransacked offices, rallied at mosque, gathered in stadium, set up roadblocks, marched through district, disrupted traffic, marched on streets, marched through rain, blocked tracks, marching through athens, gathered at port, downed tools, honked horns, threw eggs, turned out for demonstration, blocked ports, poured onto streets, gathered outside mosque, marched to hall, marched kilometres, shut down hospitals, burned effigy, set on fire, marched against summit, lit candles, converged on center, came out on streets, gathered thousands, blocked intersections, threatened strikes, picketed offices, rallied at airport, paraded through streets, stormed building, set up barricades, staged walkout, confronted police, threw stones, participated in march, marched down street, waged strike, entered day, ended strike, chanted songs, refused food, battled police, gathered outside court, gathered at park, returned to streets, pressed with strike, blocked bridge, rallied outside consulate, shaved heads, announced strike, gathered for hours, called off strike, linked hands, attacked station, erected barricades, gathered at site, converged on headquarters, intensified protests, escalated protests, rioted in streets, halted work, smashed cars, lined highway, carried streamers, hurled rocks, attacked offices, pelted embassy, staged picket, launch strike, gathered for protest, hoisted flags, threw bottles, mounted protests, surrounded headquarters, boycotted sessions, picketed embassies, pelted building, held assemblies, climbed onto roofs, occupied chamber, marched miles, thronged streets, clashed with guards, demonstrated on campus, rampaged through city, resumed strike, fought police, occupied airport, pitched tents, massed in athens left classes, marched across country, marched from office, rallied outside building, marched to border, held after prayers, organized protests, protested in paris, planned protests, gathered in cities, threw rocks, gathered at square, began boycott |

# APPENDIX D

# BOOTSTRAPPED EVENT PHRASES AND EVENT FACET PHRASES FOR THE TERRORISM DOMAIN

This appendix lists more samples from bootstrapped event dictionaries for the terrorism event domain, including event phrases and phrases for three event facets: agents, patients, and effects.

**Table D.1**: Bootstrapped Event Facet Phrase Examples for the Terrorism Domain

| |
|---|
| **Event Phrases:** claimed responsibility, hit houses, burned trucks, blew up bus, set off bomb, holding hostages, killed citizens, threatened investigators, set houses, sabotaged tower, entered residence, machinegunned residence, detonated charge, carry out attacks, attacked studios, massacred women, used bombs, maimed people, intercepted vehicles, attacking population, used explosives, unleashed escalation, stabbed them, endangering lives, expressing desire, hit helicopters, claimed credit, machinegunned car, attacked targets, placed kg, launched attacks, hurled cocktail, left leaflets, killed scores, planting bombs, taken civilians, renewed attacks, burned offices, control places, committed crime, exploded devices, set facility, planted explosives, machinegunned helicopter, machinegunned building, seized amounts, carried out murder, claimed attack, leave country, taking officials, attack students, attacked members, reiterated desire, inflicted casualties |
| **Agent Terms:** mob, ESA, group, individual, commandos, Tegucigalpa, groups, organization, forces, Willka, Montano, Santiago, organizations |
| **Patient Terms:** headquarters, citizens, officer, leaders, neighborhood, reporters, airplanes, population, Ellacuria, leader, home, buildings, office, house, residents, maids, home, leader, daughter, peasants, workers, Romero |
| **Effect Phrases:** wounded *, * be destroyed, massacre of *, broke into *, killing of *, bodies of *, * be kidnapped, * be assassinated, enter *, set fire to *, * be wounded, causing damage to *, * be shot, death of *, * be detained, machinegunned *, destroyed *, case of *, burned *, destroying *, attack on *, * be damaged, set *, killing *, leaving *, murder of *, bombed *, * be murdered, killed *, * be located, assassination of *, perpetrated against *, attacking *, release of * |

# REFERENCES


[1] ACE Evaluations, *http://www.itl.nist.gov/iad/mig/tests/ace/*.

[2] E. Agichtein and L. Gravano, *Snowball: Extracting Relations from Large Plain-Text Collections*, in Proceedings of the Fifth ACM International Conference on Digital Libraries, 2000.

[3] J. Allan, *Topic Detection and Tracking: Event Based Information Organization*, Kluwer Academic Publishers, 2002.

[4] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang, *Topic Detection and Tracking Pilot Study: Final Report*, in Proceedings of DARPA Broadcast News Transcription and Understanding Workshop, 1998.

[5] J. Allan, V. Lavrenko, and H. Jin, *First Story Detection in TDT is Hard*, in Proceedings of the International Conference on Information and Knowledge Management, 2000.

[6] J. Allan, V. Lavrenko, D. Malin, and R. Swan, *Detections, Bounds, and Timelines: Umass and TDT-3*, in Proceedings of Topic Detection and Tracking Workshop, 2000.

[7] G. Amati and C. J. Van Rijsbergen, *Probabilistic Models of Information Retrieval Based on Measuring Divergence from Randomness*, ACM Transactions on Information Systems, 20 (2002), pp. 357–389.

[8] D. Appelt, J. Hobbs, J. Bear, D. Israel, and M. Tyson, *FASTUS: a Finite-state Processor for Information Extraction from Real-world Text*, in Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI), 1993.

[9] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, *Open Information Extraction from the Web*, Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007.

[10] R. Barzilay and L. Lee, *Catching the Drift: Probabilistic Content Models, with Applications to Generation and Summarization*, in Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL 2004), 2004.

[11] H. Becker, M. Naaman, and L. Gravano, *Beyond Trending Topics: Real-world Event Identification on Twitter*, in Proceedings of the International AAAI Conference on Weblogs and Social Media, 2011.

[12] D. Beeferman, A. Berger, and J. Lafferty, *Statistical Models for Text Segmentation*, in Machine Learning, 1999.

[13] E. Benson, A. Haghighi, and R. Barzilay, *Event Discovery in Social Media Feeds*, in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-11), 2011.

[14] T. Berg-Kirkpatrick, D. Burkett, and D. Klein, *An Empirical Investigation of Statistical Significance in NLP*, in Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing, 2012.

[15] R. Bunescu and R. Mooney, *Collective Information Extraction with Relational Markov Networks*, in Proceeding of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, July 2004, pp. 438–445.

[16] ——, *Learning to Extract Relations from the Web using Minimal Supervision*, in Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, 2007.

[17] R. C. Bunescu and R. J. Mooney, *A Shortest Path Dependency Kernel for Relation Extraction*, in Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing, 2005.

[18] M. Califf and R. Mooney, *Bottom-up Relational Learning of Pattern Matching Rules for Information Extraction*, Journal of Machine Learning Research, 4 (2003), pp. 177–210.

[19] A. Carlson, J. Betteridge, E. R. Hruschka Jr., and T. Mitchell, *Coupling Semi-Supervised Learning of Categories and Relations*, in Proceedings of the NAACL HLT 2009 Workskop on Semi-supervised Learning for Natural Language Processing, 2009.

[20] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, R. Estevam, J. Hruschka, and T. Mitchell, *Toward an Architecture for Never-Ending Language Learning*, in Proceedings of the Twenty-Fourth National Conference on Artificial Intelligence, 2010.

[21] X. Carreras and L. Marquez, *Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling*, in Proceedings of the Ninth Conference on Natural Language Learning (CoNLL-2005), 2005.

[22] N. Chambers and D. Jurafsky, *Template-Based Information Extraction without the Templates*, in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-11), 2011.

[23] H. Chieu and H. Ng, *A Maximum Entropy Approach to Information Extraction from Semi-Structured and Free Text*, in Proceedings of the 18th National Conference on Artificial Intelligence, 2002.

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

[25] J. Cohen, *A Coefficient of Agreement for Nominal Scales*, in Educational and Psychological Measurement, 1960.

[26] M. D. Conover, J. Ratkiewicz, M. Francisco, B. Goncalves, A. Flammini, and F. Menczer, *Political Polarization on Twitter*, in Proceedings of the International AAAI Conference on Weblogs and Social Media, 2011.

[27] A. Culotta and J. Sorensen, *Dependency Tree Kernel for Relation Extraction*, in Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics, 2004.

[28] O. Etzioni, M. Cafarella, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates, *Unsupervised Named-Entity Extraction from the Web: An Experimental Study*, Artificial Intelligence, 165 (2005), pp. 91–134.

[29] J. Finkel, T. Grenager, and C. Manning, *Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling*, in Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, Ann Arbor, MI, June 2005, pp. 363–370.

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

[31] G. Forman, *An Extensive Empirical Study of Feature Selection Metrics for Text Classification*, J. Mach. Learn. Res., 3 (2003), pp. 1289–1305.

[32] D. Freitag, *Multistrategy Learning for Information Extraction*, in Proceedings of the Fifteenth International Conference on Machine Learning, Morgan Kaufmann Publishers, 1998.

[33] ——, *Toward General-Purpose Learning for Information Extraction*, in Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics, 1998.

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

[35] D. Gildea and D. Jurafsky, *Automatic Labeling of Semantic Roles*, Computational Linguistics, 28 (2002), pp. 245–288.

[36] D. Grenon and B. Smith, *SNAP and SPAN: towards Dynamic Spatial Ontology*, (2004), pp. 69–104.

[37] Z. Gu and N. Cercone, *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, Sydney, Australia, July 2006, pp. 481–488.

[38] A. Haghighi, K. Toutanova, and C. Manning, *A Joint Model for Semantic Role Labeling*, in Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL), 2005, pp. 173–176.

[39] M. Hearst, *TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages*, in Computational Linguistics, 1997.

[40] ——, *Design Recommendations for Hierarchical Faceted Search Interfaces*, in ACM SIGIR Workshop on Faceted Search, 2006.

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

[42] J. Hobbs and E. Riloff, *Information Extraction*, in Handbook of Natural Language Processing, 2nd Edition, 2010.

[43] R. Hoffmann, C. Zhang, and D. Weld, *Learning 5000 Relational Extractors*, in Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 2010.

[44] R. Huang and E. Riloff, *Peeling Back the Layers: Detecting Event Role Fillers in Secondary Contexts*, in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-11), 2011.

[45] ——, *Modeling Textual Cohesion for Event Extraction*, in Proceedings of the 26th Conference on Artificial Intelligence (AAAI-12), 2012.

[46] ——, *Multi-faceted Event Recognition with Bootstrapped Dictionaries*, in Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-13), 2013.

[47] S. Huffman, *Learning Information Extraction Patterns from Examples*, in Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing, S. Wermter, E. Riloff, and G. Scheler, eds., Springer-Verlag, Berlin, 1996, pp. 246–260.

[48] H. Ji and R. Grishman, *Refining Event Extraction through Cross-Document Inference*, in Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08), Columbus, OH, June 2008, pp. 254–262.

[49] X. Ji and H. Zha, *Domain-independent Text Segmentation Using Anisotropic Diffusion and Dynamic Programming*, in Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2003.

[50] H. Jin, R. Schwartz, S. Sista, and F. Walls, *Topic Tracking for Radio, TV Broadcast, and Newswire*, in Proceedings of European Conference on Speech Communication and Technology, 1999.

[51] T. Joachims, *Making Large-Scale Support Vector Machine Learning Practical*, in Advances in Kernel Methods: Support Vector Machines, A. S. B. Schölkopf, C. Burges, ed., MIT Press, Cambridge, MA, 1999.

[52] T. Joachims, *Transductive Inference for Text Classification using Support Vector Machines*, in Proceedings of International Conference on Machine Learning, 1999.

[53] K. Kaneiwa, M. Iwazume, and K. Fukuda, *An Upper Ontology for Event Classifications and Relations*, in Proceedings of the 20th Australian Joint Conference on Advances in Artificial Intelligence, 2007.

[54] A. Kazantseva and S. Szpakowicz, *Linear Text Segmentation Using Affinity Propagation*, in Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, 2011.

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

[56] A. Kehagias, P. Fragkou, and V. Petridis, *Linear Text Segmentation Using a Dynamic Programming Algorithm*, in Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics, 2003.

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

[58] J. Kim, T. Ohta, S. Pyysalo, Y. Kano, and J. Tsujii, *Overview of the BioNLP09 Shared Task on Event Extraction*, in Proceedings of the NAACL-HLT 2009 Workshop on Natural Language Processing in Biomedicine (BioNLP09), 2009.

[59] J. Lafferty, A. McCallum, and F. Pereira, *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*, in Proceedings of the Eighteenth International Conference on Machine Learning, 2001.

[60] V. Lampos, T. D. Bie, and N. Cristianini, *Flu Detector - Tracking Epidemics on Twitter*, in ECML PKDD, 2010.

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

[62] Q. Li, H. Ji, and L. Huang, *Joint Event Extraction via Structured Prediction with Global Features*, in Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-13), 2013.

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

[64] S. Liao and R. Grishman, *Using Document Level Cross-Event Inference to Improve Event Extraction*, in Proceedings of the 48st Annual Meeting on Association for Computational Linguistics (ACL-10), 2010.

[65] Z. Lin, H. T. Ng, and M.-Y. Kan, *A PDTB-Styled End-to-End Discourse Parser*, in Technical Report TRB8/10, National University of Singapore, August, 2010.

[66] W. Lu and D. Roth, *Automatic Event Extraction with Structured Preference Modeling*, in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, 2012.

[67] I. Malioutov and R. Barzilay, *Minimum Cut Model for Spoken Lecture Segmentation*, in Proceedings of Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING/ACL-06), 2006.

[68] M.-C. d. Marneffe, B. MacCartney, and C. D. Manning, *Generating Typed Dependency Parses from Phrase Structure Parses*, in Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC-2006), 2006.

[69] M. Maslennikov and T. Chua, *A Multi-Resolution Framework for Information Extraction from Free Text*, in Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, 2007.

[70] M. Mathioudakis and N. Koudas, *TwitterMonitor: Trend Detection over the Twitter Stream*, in Proceedings of the 2010 international conference on Management of data, ACM, 2010, p. 11551158.

[71] A. McCallum, D. Freitag, and F. Pereira, *Maximum Entropy Markov Models for Information Extraction and Segmentation*, in Proceedings of the Seventeenth International Conference on Machine Learning, 2000.

[72] D. McClosky, M. Surdeanu, and C. Manning, *Event Extraction as Dependency Parsing for BioNLP 2011*, in Proceedings of the BioNLP 2011 Workshop Companion Volume for Shared Task, 2011.

[73] D. Metzler, C. Cai, and E. Hovy, *Structured Event Retrieval over Microblog Archives*, in Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-12), 2012.

[74] G. Miller, *Wordnet: An On-line Lexical Database*, International Journal of Lexicography, 3 (1990).

[75] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, *Distant supervision for Relation Extraction without Labeled Data*, in Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-09), 2009.

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

[77] MUC-5 Proceedings, *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, 1993.

[78] MUC-6 Proceedings, *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, 1995.

[79] MUC-7 Proceedings, *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998.

[80] K. Nanda, *Combining Lexical, Syntactic and Semantic Features with Maximum Entropy Models for Extracting Relations*, in Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics, 2004.

[81] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, *Text Classification from Labeled and Unlabeled Documents using EM*, Machine Learning, 39 (2000), pp. 103–134.

[82] I. Ounis, C. Lioma, C. Macdonald, and V. Plachouras, *Research Directions in Terrier*, Novatica/UPGRADE Special Issue on Web Information Access, Ricardo Baeza-Yates et al. (Eds), Invited Paper, (2007).

[83] R. Parker, D. Graff, J. Kong, K. Chen, and K. Maeda, *English Gigaword*, in Linguistic Data Consortium, 2011.

[84] S. Patwardhan, *Widening the Field of View of Information Extraction through Sentential Event Recognition*, in Ph.D. Dissertation, 2010.

[85] S. Patwardhan and E. Riloff, *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), 2007.

[86] F. Peng and A. McCallum, *Accurate Information Extraction from Research Papers using Conditional Random Fields*, in Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL 2004), 2004.

[87] S. Petrovic, M. Osborne, and V. Lavrenko, *Using Paraphrases for Improving First Story Detection in News and Twitter*, in Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-12), 2012.

[88] A.-M. Popescu, M. Pennacchiotti, and D. A. Paranjpe, *Extracting Events and Event Descriptions from Twitter*, in Proceedings of the 20th International World Wide Web Conference (WWW 2011), 2011.

[89] M. Porter, *An Algorithm for Suffix Stripping*, Program, 14 (1980), pp. 130–137.

[90] R. Prasad, N. Dinesh, L. A., E. Miltsakaki, L. Robaldo, J. A., and B. Webber, *The Penn Discourse Treebank 2.0*, in Proceedings of the Sixth Conference on Language Resources and Evaluation (LREC-2008), 2008.

[91] ProMed-mail, *http://www.promedmail.org/*, 2006.

[92] V. Punyakanok, D. Roth, W. Yih, D. Zimak, and Y. Tu, *Semantic Role Labeling via Generalized Inference over Classifiers (Shared Task Paper)*, in Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL), H. T. Ng and E. Riloff, eds., 2004, pp. 130–133.

[93] L. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of the IEEE, 77 (1989), pp. 257–286.

[94] E. Riloff, *Automatically Constructing a Dictionary for Information Extraction Tasks*, in Proceedings of the 11th National Conference on Artificial Intelligence, 1993.

[95] ——, *Automatically Generating Extraction Patterns from Untagged Text*, in Proceedings of the Thirteenth National Conference on Artificial Intelligence, The AAAI Press/MIT Press, 1996, pp. 1044–1049.

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

[97] E. Riloff and W. Lehnert, *Information Extraction as a Basis for High-Precision Text Classification*, ACM Transactions on Information Systems, 12 (1994), pp. 296–333.

[98] E. Riloff and J. Lorenzen, *Extraction-based Text Categorization: Generating Domain-specific Role Relationships Automatically*, in Natural Language Information Retrieval, T. Strzalkowski, ed., Kluwer Academic Publishers, 1999.

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

[100] A. Ritter, Mausam, O. Etzioni, and S. Clark, *Open Domain Event Extraction from Twitter*, in The 18th ACM SIGKDD Knowledge Discovery and Data Mining Conference, 2012.

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

[102] P. S. and R. E., *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), 2009.

[103] T. Sakaki, M. Okazaki, and Y. Matsuo, *Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors*, in Proceedings of the 19th International World Wide Web Conference (WWW 2010), 2010.

[104] C. Sauper, A. Haghighi, and R. Barzilay, *Incorporating Content Structure into Text Analysis Applications*, in Proceedings of 2010 the Conference on Empirical Methods in Natural Language Processing (EMNLP-2010), 2010.

[105] S. Sekine, *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), 2006.

[106] Y. Shinyama and S. Sekine, *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, New York City, NY, June 2006, pp. 304–311.

[107] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert, *CRYSTAL: Inducing a Conceptual Dictionary*, in Proc. of the Fourteenth International Joint Conference on Artificial Intelligence, 1995, pp. 1314–1319.

[108] S. Soderland and W. Lehnert, *Wrap-Up: A Trainable Discourse Module for Information Extraction*, Journal of Artificial Intelligence Research (JAIR), 2 (1994), pp. 131–158.

[109] R. Soricut and D. Marcu, *Discourse Generation Using Utility-Trained Coherence Models*, in Proceedings of Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING/ACL-06), 2006, pp. 803–810.

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

[111] K. Sudo, S. Sekine, and R. Grishman, *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), 2003.

[112] C. Thompson, R. Levy, and C. D. Manning, *A Generative Model for Semantic Role Labeling*, in In Proceeding of 14th European Conference on Machine Learning, 2003.

[113] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, *Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment*, in Proceedings of the International AAAI Conference on Weblogs and Social Media, 2010.

[114] D. Tunkelang, *Faceted Search*, Morgan and Claypool Publishers, 2009.

[115] M. Worboys and K. Hornsby, *From Objects to Events: GEM, the Geospatial Event Mode*.

[116] N. Xue and M. Palmer, *Calibrating Features for Semantic Role Labeling*, in In Proceedings of the Conference on Empirical Methos in Natural Language Processing, 2004, pp. 88–94.

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

[118] L. Yao, A. Haghighi, S. Riedel, and A. McCallum, *Structured Relation Discovery using Generative Models*, in Proceedings of 2011 the Conference on Empirical Methods in Natural Language Processing (EMNLP-2011), 2011.

[119] S. Yi and M. Palmer, *The Integration of Syntactic Parsing and Semantic Role Labeling*, in Proceedings of the Ninth Conference on Natural Language Learning (CoNLL-2005), 2005.

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

[121] D. Zelenko, C. Aone, and A. Richardella, *Kernel Methods for Relation Extraction*, Journal of Machine Learning Research, 3 (2003).

[122] M. Zhang, J. Zhang, and J. Su, *Exploring Syntactic Features for Relation Extraction using a Convolution Tree Kernel*, in Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, 2006.

[123] M. Zhang, J. Zhang, J. Su, and G. Zhou, *A Composite Kernel to Extract Relations between Entities with both Flat and Structured Features*, in Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association of Computational Linguistics (COLING/ACL-2006), 2006.

[124] X. Zhang, H. Fuehres, and P. A. Gloor, *Predicting Stock Market Indicators through Twitter "I hope it is not as bad as I fear"*, in COINs, 2010.

[125] G. Zhou, J. Su, J. Zhang, and M. Zhang, *Exploring Various Knowledge in Relation Extraction*, in Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics, 2005.

[126] G. Zhou, M. Zhang, D. Ji, and Q. Zhu, *Tree Kernel-based Relation Extraction with Context-Sensitive Structured Parse Tree Information*, in Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, 2007.