WIDENING THE FIELD OF VIEW OF INFORMATION EXTRACTION THROUGH SENTENTIAL EVENT RECOGNITION

by

Siddharth Patwardhan

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

May 2010

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

SUPERVISORY COMMITTEE APPROVAL

of a dissertation submitted by

Siddharth Patwardhan

This dissertation has been read by each member of the following supervisory committee and by majority vote has been found to be satisfactory.

	Chair: Ellen Riloff
	Hal Daumé III
	John Hurdle
	Gary Lindstrom
	Janyce Wiebe

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

FINAL READING APPROVAL

To the Graduate Council of the University of Utah:

I have read the dissertation of Siddharth Patwardhan in its final form and have found that (1) its format, citations, and bibliographic style are consistent and acceptable; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the Supervisory Committee and is ready for submission to The Graduate School.

Date

Ellen Riloff
Chair, Supervisory Committee

Approved for the Major Department

Martin Berzins
Chair/Dean

Approved for the Graduate Council

Charles A. Wight
Dean of The Graduate School

ABSTRACT

Event-based Information Extraction (IE) is the task of identifying entities that play specific roles within an event described in free text. For example, given text documents containing descriptions of disease outbreak events, the goal of an IE system is to extract event role fillers, such as the disease, the victims, the location, the date, etc., of each disease outbreak described within the documents. IE systems typically rely on local clues around each phrase to identify their role within a relevant event. This research aims to improve IE performance by incorporating evidence from the wider sentential context to enable the IE model to make better decisions when faced with weak local contextual clues.

To make better inferences about event role fillers, this research introduces an "event recognition" phase, which is used in combination with localized text extraction. The event recognizer operates on sentences and locates those sentences that discuss events of interest. Localized text extraction can then capitalize on this information and identify event role fillers even when the evidence in their local context is weak or inconclusive. First, this research presents Piper, a pipelined approach for IE incorporating this idea. This model uses a classifier-based sentential event recognizer, combined with a pattern-based localized text extraction component, cascaded in a pipeline. This enables the pattern-based system to exploit sentential information for better IE coverage. Second, a unified probabilistic approach for IE, called Glacier, is introduced to overcome limitations from the discrete nature of the pipelined model. Glacier combines the probability of event sentences, with the probability of phrasal event role fillers into a single joint probability, which helps to better balance the influence of the two components in the IE model. An empirical evaluation of these models shows that the use of an event recognition phase improves IE performances, and it shows that incorporating such additional information through a unified probabilistic model produces the most effective IE system.

To Navdeep

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Many people have had an enormous influence in making this dissertation a reality. While it is impossible to mention everyone by name, I make an attempt here to acknowledge my gratitude to those who have had a hand in the success of this work.

I am eternally grateful to my advisor, Dr. Ellen Riloff, for her guidance, her support and most of all, her patience throughout this research. I have learned so much from her and it has truly been a priviledge to work with her.

I would also like to thank all of my committee members, Dr. Hal Daumé, Dr. John Hurdle, Dr. Gary Lindstrom and Dr. Janyce Wiebe, all of whom have been instrumental in making this work a success. I am indebted to each one of them for their valuable discussions and input at every step in my dissertation research. Additionally, my gratitude also goes out to Dr. Al Davis for his guidance on my committee during the early stages of my research.

Many thanks to my colleagues, Nathan Gilbert and Adam Teichert, for their help with the annotation study conducted in this dissertation. Needless to say, without their patience and hard work, this dissertation would have been incomplete.

I learned so much about research and about NLP from my MS thesis advisor, Dr. Ted Pedersen, and I will always be grateful to him for introducing me to the field and for teaching me to become an effective researcher.

Ideas in this work were greatly influenced by many useful discussions with friends and colleagues in the NLP Research Group at the University of Minnesota Duluth as well as the NLP Lab here at the University of Utah. Special thanks to Bill Phillips, Sean Igo, David Price, Carolin Arnold, Brijesh Garabadu, Amit Goyal, Arvind Agarwal, Piyush Rai, Ruihong Huang, Amrish Kapoor, Satanjeev Banerjee, Amruta Purandare, Dr. Saif Mohammad and Dr. Bridget McInnes for their support and for fostering an environment encouraging discussions and ideas in our graduate research labs.

During the course of this research I had the wonderful opportunity of presenting my research work at the AAAI/SIGART Doctoral Consortium at the AAAI 2008 conference. I am indebted to Dr. Kiri Wagstaff for mentoring me at this event, to AAAI/SIGART for the opportunity and to all of the participants at the Doctoral Consortium for their valuable feedback.

I have had the good fortune of collaborating with many smart folks over the years, and these collaborations have had a significant positive influence on my understanding of NLP. I am grateful to my collaborators for broadening my horizons in the world of NLP. I am also grateful to Dr. Steve Gates, Dr. Youngja Park, Dr. Serguei Pakhomov and Dr. James Buntrock for mentoring me during summer internships at the IBM T. J. Watson Research Center and at the Mayo Clinic.

These acknowledgments would be incomplete without mentioning the immense support of the staff at the School of Computing in the University of Utah. Special thanks go to Karen Feinauer, whose efficiency is unparalleled. She always made sure that I was on track, meeting all University deadlines and requirements, and her help with this over these past years has been invaluable.

My wife Navdeep, my brother Gautam and my parents Ajeet and Alka have always supported me in everything I have chosen to do. They have provided me with the much needed emotional support that kept me sane during this research. This work would have been impossible to complete in the absence of this much needed support.

Last, but not the least, I would like to acknowledge the various funding agencies that have supported my work over the years: the Advanced Research and Development Activity (ARDA), the National Science Foundation (NSF), the Institute for Scientific Computing Research and the Center for Applied Scientific Computing within Lawrence Livermore National Laboratory (LLNL) and the Department of Homeland Security (DHS).

CHAPTER 1

INTRODUCTION

With steady progress in natural language technology in recent years, there has been a growing interest in the use of text analysis systems in everyday applications. Government agencies, research institutions and corporations are all realizing the value of information in free text, especially as technology makes access to such information more feasible. We can already find numerous examples of this around us. Medical institutions are looking to use the vast quantities of patient records generated every day to automatically detect and prevent unexpected, harmful or fatal drug reactions [7, 72, 85]. Similarly, government agencies are funding research for automatically detecting and tracking disease outbreaks described in online texts and public mailing lists [44, 43, 82]. Textual content analysis is also being widely used to gauge the opinions of people on commercial products and popular media [80, 51, 88, 120]. All of these applications automatically identify useful information within unstructured text, making analysis tasks in various fields easier for humans.

There are many different types of useful information that exist in free text. Several text processing tasks have been specifically defined with the goal of locating such information in text. These include recognizing named entities [23, 75, 14, 54, 59], extracting relations [129, 27, 9, 10], extracting information about events and entities from semistructured text [37, 18, 84, 42, 126], and from free text [92, 124, 15, 69]. The type of information located by an automated system is usually determined by the end application or human expert using that information. The useful information found in text is then typically organized into structured representations, such as templates or database entries, making it easier to further analyze or process this information. Other applications can utilize this structured representation of textual information in data analysis projects that help improve the quality of our lives.

Experience has shown that computers are much better at processing structured information than they are at processing unstructured information. This is the primary motivation for designing Information Extraction (IE) systems. The goal of such systems is to locate text spans representing information of interest in free text. The structured representation

of text generated by these systems enables various forms of data analysis, which would be impossible to do with free text. For example, disease outbreak information extracted by such systems [44, 43, 82] could be used to automatically track the spread of specific diseases and to take measures to cease their escalation. Similarly, systems could be built to extract facts about famous people or locations (e.g., birthdays of celebrities, authors of books, capitals of countries or states). In addition, such systems have been used as essential components in other important Natural Language Processing (NLP) applications such as *Question Answering* [108, 90, 109, 93], *Subjectivity Analysis* [16, 94], etc. Thus, IE is an important piece of technology, which can become the basis of various types of data analysis and applications.

In particular, many applications or analysts depend on information about specific events. For instance, a stock market analyst would be interested in timely information about any major events associated with companies. An IE system could assist such an analyst by automatically locating and summarizing these events, such as mergers and acquisitions of corporations, or changes in the management of companies [105, 39, 124]. Knowing about the roles played by various entities within the events is of particular interest in such cases. The goal of IE systems for this particular type of task is, therefore, to identify specific events of interest in given text and, additionally, identify the various entities (which could be people, locations, objects, etc.) that play specific roles (e.g., a company being acquired) within the detected events.

The focus of this dissertation is on the study of techniques for locating information about events described in free text. The dissertation argues that, to identify the roles played by entities in events of interest, inferences based on evidence in the wider context are required. For an automated system to extract such information, it must be able to exploit such wider contextual evidence in a similar manner. The research presented here shows how IE performance can be improved by detecting relevant event descriptions in the sentential context, and augmenting the local phrasal evidence with this sentential event information.

The remainder of this chapter describes more clearly the task at hand, provides an overview of previous approaches for this task, and presents an outline of novel techniques presented in this dissertation to improve upon previous approaches. The claims and scientific contributions of this research are then enumerated in Section 1.4. Finally, to guide the reader through the rest of this dissertation a brief overview of each of the remaining chapters is presented in Section 1.5.

1.1 Event-based Information Extraction

The focus of the research presented in this dissertation is on "event-based" *Information Extraction* (IE), which is the task of extracting pieces of information pertaining to specific events from free text. The objective is to automatically locate descriptions of certain events within the given text, and for each event extract the mentions of various individuals, objects, dates and locations that play specific roles within the event. The events of interest, as well as the roles of interest, are defined beforehand — typically in accordance with the needs of an end application or authority utilizing this information. To elaborate, consider a hypothetical example, where an organization such as the Centers for Disease Control and Prevention (CDC) is interested in gathering statistics about disease outbreaks described in numerous reports and text documents sent to them. In such a situation, the CDC specifies that disease outbreak events as being the events of interest for the IE task. Further for each outbreak event, they would like to know the name of the disease, the list of victims, the location of the event, and other such information pertaining to the event. This event information is defined as a set of roles, or what we will refer to as *event roles* henceforth in this document. Figure 1.1 illustrates the expected outcome for the given IE task on a sample document. The event roles presented in this example are *disease*, *victims*, *location*, *date* and *status*. Observe that event-based IE systems, in essence, process unstructured textual information and convert it to a more structured representation, viz. *event templates* or *database entries*.

Although event-based IE may appear to be an easy task for humans to accomplish, it presents numerous challenges to automated systems. For one, the information to be extracted is often sparsely distributed within the given texts. The victim of a disease outbreak event, for instance, may be mentioned just once in an entire document, and the IE system must find and extract this one word or phrase from the text. Indeed, some researchers [42] have referred to this as a "nugget extraction" task for this very reason. Furthermore, there may be multiple events described in a document. There may also be no events described within a document. An event description may have multiple event roles

New Jersey, February, 26. An outbreak of Ebola has been confirmed in Mercer County, NJ. Five teenage boys appear to have contracted the deadly virus from an unknown source. The CDC is investigating the cases and is taking measures to prevent the spread...

Disease:	*Ebola*
Victims:	*Five teenage boys*
Location:	*Mercer County, NJ*
Date:	*February 26*
Status:	*Confirmed*

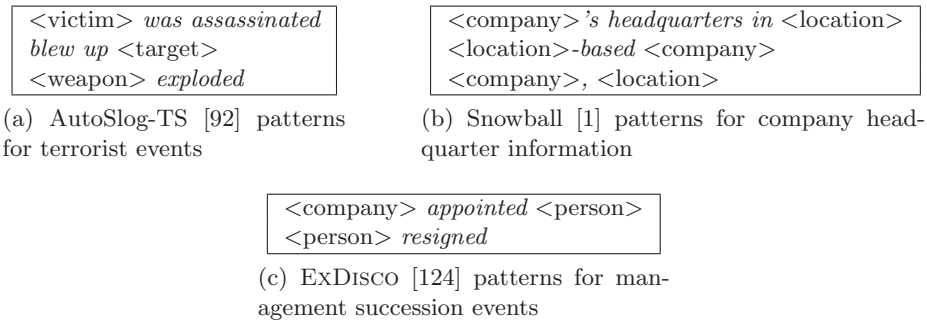
Figure 1.1: Example *disease outbreak* event template generated from free text

that need to be extracted, which makes matters more complex for an IE system. In addition to these complexities, the nature of event descriptions also varies widely. Sometimes, events may be described in great detail across the entire document, while sometimes there may only be a fleeting reference to an event within the document. In addition, some documents may contain short summaries of multiple events. All of these variations in event descriptions, and in the roles played by entities within these events, make event-based IE a significantly challenging task.

Despite these numerous challenges, the benefits of structured representation generated during IE have prompted researchers to explore techniques for automatically and accurately extracting event-role information from free text. In some of the early work on event-based IE, researchers observed that contextual information surrounding a word or a phrase in text was frequently a strong indicator of the role played by that word or phrase in an event of interest. For instance, the subject of the passive voice verb phrase "*was assassinated*" is usually the victim of a criminal act. Thus, searching for this specific "pattern" in text can very accurately extract victims of crimes. Similarly, extracting the noun phrase appearing as the object of the preposition "of" in "*outbreak of*" is usually the disease in a disease outbreak event. From observations such as these, the use of "extraction patterns" representing phrasal context developed as a common approach for information extraction.

Extraction patterns define constraints on lexical, syntactic or semantic properties of text segments. These patterns are applied to text documents for locating relevant information. When applied to text, if the constraints defined by a pattern are met, then a portion of the text (typically, a word or a phrase) is extracted by the pattern. For example, a pattern such as "<subject> *was assassinated*" defines a lexical constraint on verb phrases in text. If the head of a verb phrase is the verb "*assassinated*" in passive voice, then the constraints defined by the pattern are met, and the subject of that verb phrase is extracted by the pattern.

Early approaches to IE used hand-crafted patterns for event role extraction. More recently, the goal of pattern-based IE approaches (e.g., [124, 92, 1, 12, 17]) has been to learn a set of extraction patterns for a given IE task (i.e., for a specific type of event). All of these approaches emphasize a syntactic analysis of the text (part-of-speech tagging, dependency parsing, chunking, etc.), and the patterns used in these approaches are typically built upon such textual analysis. Figure 1.2 shows examples of extraction patterns learned by some pattern-based IE systems, where the event roles defined within < and > are extracted by the patterns. The key challenge for such pattern-based systems is the learning



(a) AutoSlog-TS [92] patterns for terrorist events

(b) Snowball [1] patterns for company headquarter information

(c) ExDisco [124] patterns for management succession events

Figure 1.2: Examples of extraction patterns used by some IE systems

strategy used to generate a set of good extraction patterns for the given event of interest. Broadly, the current approaches for learning such patterns include bootstrapping techniques, weakly supervised learning, fully supervised learning, among others. The patterns learned from these approaches are then used to extract event information from new (unseen) text documents.

More recently, the trend has been to employ machine learning classifiers to make extraction decisions about words or phrases in text documents. These approaches (e.g., [38, 39, 15, 8, 36]) view the input text as a stream of tokens or phrases, and use features from the contexts around them to decide if they should be extracted. The features used by these classifiers include lexical, syntactic and semantic properties of text in the context surrounding each word or phrase. Thus, just like the pattern-based approaches, these classifier-based approaches also primarily rely on features of the local context around words or phrases in making their extraction decisions. A more detailed overview of current approaches, pattern-based as well as classifier-based, appears in Chapter 2, which further emphasizes the dependence of these approaches on local contextual information.

1.2 Motivation

As illustrated in the previous section, current approaches for IE decide which words in the given text should be extracted primarily by analyzing their contexts. However, most IE approaches are designed to focus only on a small window of context surrounding each potential extraction, before making a decision. The reason for focusing on a small window of context is that as the window of context increases, the patterns matching the larger contexts appear less frequently in a corpus of text. This sparse nature of larger contexts and their corresponding patterns makes them less useful as indicators of relevant extractions. For example, a larger context such as "*the yellow house with the large backyard*" is less likely

to appear in a corpus of text than a smaller context such as "*the yellow house*." Thus, it is easier to gather statistics for evaluating patterns representing shorter contexts. As shown by the examples in Figure 1.2, extraction patterns are designed to match only a small window of context surrounding the extraction. Indeed, most pattern-based approaches typically learn patterns designed to match an extremely local context (one to six words) around the potential extractions.

The problem with focusing on a limited context during extraction is that, many times, the small context does not contain sufficient evidence to definitively conclude if a certain piece of text should be extracted. For example, Figure 1.3a presents a small news clipping[1] describing a recent terrorist event in Iraq. The weapon used in this terrorist event is "*rocket or mortar ammunition*." However, observe that the local context ("*Maj. Brad Leighton described as* <weapon>") around this span of text provides almost no evidence of a terrorist event. Consequently, an IE approach relying solely on this local context would be unlikely to extract any information from this span of text.

Likewise, current techniques tend to extract erroneous information in another completely different type of situation — a *misleading local context*. Very often, because of idiomatic or metaphorical use of language, the local context around a word or phrase may strongly indicate an event role extraction, when in fact the wider context (the sentence or the paragraph) proves otherwise. To prevent incorrect extractions in such cases, we need the evidence from the wider context to override the local contextual evidence. For example,

Rebels fired an explosive barrage yesterday into the capital's protected Green Zone, targeting the heart of America's diplomatic and military mission in Iraq. There were no injuries. The strikes were the most recent involving what U.S. Maj. Brad Leighton described as rocket or mortar ammunition.

(a) Example illustrating relevant information embedded in a context that is not event-specific

President Barack Obama was attacked from all sides on Friday over his decision to declassify four memos detailing harsh CIA interrogation methods approved by the George W. Bush administration for use against terror suspects. Former senior Bush officials criticised the president for giving away secrets to terrorists, and claimed that the tactics had worked.

(b) Example illustrating irrelevant information embedded in seemingly relevant context

Figure 1.3: News snippets illustrating deficiencies of current IE approaches

[1]Quoted from *The Edmonton Sun* online edition (http://www.edmontonsun.com), February 24, 2008.

consider the news clipping[2] in Figure 1.3b. Locally, a pattern such as "<subject> *was attacked*" seems likely to identify the victim of a physical attack. But from the wider context we see that it is not a physical attack at all. Indeed, evidence from the wider context can help to reveal such false positives.

Let us further explore the limitations of local context in locating event roles of an event. Most IE approaches are designed to identify role fillers that appear as arguments to event verbs or nouns, either explicitly via syntactic relations or implicitly via proximity (e.g., *John murdered Tom* or *the murder of Tom by John*). But many facts are presented in clauses that do not contain event words, requiring discourse relations or deep structural analysis to associate the facts with event roles. Consider the sentences in Figure 1.4, for example, which illustrate a common phenomenon in text. In these examples information is not explicitly stated as filling an event role, but human readers have no trouble making this inference. The role fillers above (***seven people***, ***two bridges***) occur as arguments to verbs that reveal state information (death, destruction) but are not event-specific (i.e., death and destruction can result from a wide variety of incident types). IE approaches often fail to extract these role fillers because they do not recognize the immediate context as being relevant to the specific type of event they are looking for.

Models for extraction that rely primarily on the local context for IE do well on cases that contain direct evidence of the event and the event role within their local context (e.g., *the chairman was assassinated*). However, as seen in the examples above, these models would fail in cases where the reader must make inferences based on information from a

Seven people *have died*

. . . in Mexico City and its surrounding suburbs in a Swine Flu outbreak.

. . . after a tractor-trailer collided with a bus in Arkansas.

. . . and 30 were injured in India after terrorists launched an attack on the Taj Mahal Hotel yesterday.

Two bridges *were destroyed*

. . . to make way for modern, safer bridges to be constructed early next year.

. . . in Baghdad last night in a resurgence of bomb attacks in the capital city.

. . . and $50 million in damage was caused by a hurricane that hit Miami on Friday.

Figure 1.4: Examples illustrating the need for wider contextual evidence

[2]Quoted from *The Telegraph* online edition (http://www.telegraph.co.uk), April 17, 2009.

wider context. We can, therefore, hope for improvement in IE performance if the model could account for this global information from the wider context in its extraction decisions. The goal of this dissertation is to study the use of event information in the wider context for better IE performance.

An observation in existing pattern-based IE systems that reinforces the need for wider contextual evidence is that even apparently reliable patterns learned by these systems may not always reperesent an event-specific context. For instance, one would expect a pattern like "<victim> *was kidnapped*" learned by a pattern-based IE system (AutoSlog-TS [92]) to be a strong indicator of terrorist events, and a good victim extractor. But when applied to text documents, this pattern does not always identify relevant information. On the MUC-4 data set [114], it extracts a correct answer 83% of the time. Similarly, patterns like "*murder of* <victim>" and "<victim> *was killed*" achieve a precision of 65% and 31%, respectively. The lower precision occurs as a result of false hits in nonevent contexts. The patterns could, therefore, benefit from additional event information from the wider context. Knowing when a pattern appears in a nonevent sentence or paragraph can help improve its precision by preventing extractions in such cases. Thus, such evidence from the wider context can benefit even existing pattern-based IE approaches.

Indeed, many recent approaches have begun exploring techniques for IE that incorporate global information from text to help improve extraction performance. Maslennikov and Chua [69] use discourse trees and local syntactic dependencies in a pattern-based framework to incorporate wider context. Finkel et al. [36] and Ji and Grishman [55] incorporate global information by enforcing event role or label consistency over a document or across related documents. These techniques have all shown the benefits of global information for event-based IE. The research presented in this dissertation takes an *event detection* approach for improving IE with evidence from the sentential context.

1.3 Overview of the Research

Clearly, from the previous discussion, we see that local contextual information surrounding a word or phrase is frequently insufficient for deciding its role in an event. This dissertation takes an event detection approach for incorporating evidence from a wider context. The idea is to build a model to automatically detect event descriptions in text.

The knowledge of an event description within text or a conversation enables human readers or listeners to make several assumptions about the entities or objects under discussion. These assumptions are typically based on our prior knowledge of the world. Thus, conversations and text documents typically omit very specific details from the exposition,

and these are left for the reader or the listener to infer. Figure 1.5, for example, contains two sentences (taken from a MUC-4 document) describing a terrorist attack on Attorney General Garcia Alvarado. The text provides a description of the attack, but leaves a lot to the imagination of the reader. Three facts are explicitly stated in the text:

(a) an attack took place,

(b) the Attorney General's car stopped at a traffic light, and

(c) an individual placed a bomb on the roof of the armored vehicle

From just these three facts we make several inferences, and reconstruct the event as we imagine it might have occurred. We infer that perhaps:

(a) the Attorney General's car is the armored vehicle,

(b) the Attorney General was in the car at the time, and

(c) the bomb subsequently exploded

This ultimately leads us to conclude that the Attorney General was the victim of the attack, and the bomb was the weapon used in this terrorist attack. Even though many details of the attack have not been explicitly stated in the text, we (the human readers) effortlessly infer these from the context. The victim and weapon of this terrorist event are deduced from this text with no direct evidence of these event roles in the local context of the phrases.

An analogy can be drawn from the field of Computer Vision. One of the tasks for computers analyzing images and photographs is to try to recognize individual objects in the given picture. It is usually harder to recognize certain objects devoid of any global context. Conversely, it is easier to recognize objects, even without physical and textural details, when presented in the context of a scene in which it appears. For example, given the scene of a road, a human observer would usually infer that a red object on the road is a car, unless the physical details of the car suggest otherwise. Thus, the global scene plays an important role in object recognition in the human visual faculty. Figure 1.6 illustrates this phenomenon with two images, one containing an object without context, and another with

ACCORDING TO CRISTIANI, THE ATTACK TOOK PLACE BECAUSE ATTORNEY GENERAL GARCIA ALVARADO WARNED THAT "HE WOULD TAKE MEASURES AGAINST URBAN TERRORISTS."

VICE PRESIDENT-ELECT FRANCISCO MERINO SAID THAT WHEN THE ATTORNEY GENERAL'S CAR STOPPED AT A LIGHT ON A STREET IN DOWNTOWN SAN SALVADOR, AN INDIVIDUAL PLACED A **BOMB** ON THE ROOF OF THE ARMORED VEHICLE.

Figure 1.5: Excerpt from MUC-4 data [114]: extractions from nonevent contexts

(a) Object without context (b) Object within context of scene

Figure 1.6: Images illustrating the significance of context in object recognition

an object in the context of a scene with other objects. To recognize the object in Figure 1.6a we only have the benefit of the visual features of the object, which do not conclusively enable us to recognize the object shown. Figure 1.6b, on the other hand, presents exactly the same object in its natural surroundings — on the desktop. This context allows us to make contextual inferences about the object and recognize it as the mouse used in a computer desktop system. Many of the earlier approaches for automatic object recognition relied solely on specific properties of objects to recognize them. More recently, however, researchers have had success by accounting for the global context within which the object appears. Torralba et al. [118] show how recognizing the scene (office, street, corridor, etc.) in which an object appears helps provide "contextual priors" for better object recognition. Similarly, Oliva and Torralba [79] illustrate how a statistical summary of a scene enables contextual inference in recognizing individual objects in the scene. A similar phenomenon is seen in text, where the description of the scene or the event being discussed allows us to make contextual inferences about various entities discussed in the text.

To enable inferences based on the scene or event description in text, this research presents a two-stage approach for IE. The models for IE described here contain two primary components — an event recognition component to capture the essence of the scene described in text, and a local component to identify roles played by entities in an event. This dissertation starts with a description of a "pipelined" two-stage model for IE called Piper, which first identifies event sentences within the given text, and then applies a localized extraction model to take advantage of the identified sentences. Further building upon the Piper implementation, this dissertation then follows up with a more flexible model for IE called Glacier. This model employs a unified approach that probabilistically

combines information from the two components without making discrete decisions within the individual components. The IE models presented in this dissertation set forth two different strategies for incorporating information from a wider context to enable the extraction of event roles that have no direct evidence of a relevant event in their immediate context.

PIPER, the pipelined model for IE, consists of two stages in a pipeline running independently of one another. Figure 1.7 presents a block schematic of this model. As shown in the figure, the *sentential event recognizer* is first applied to the input documents, and detects sentences that contain the description of a relevant event. This component uses text classification techniques for identifying event sentences. Having detected event sentences in the input text, the *localized text extraction* component then extracts event roles from these sentences. This component of the model can now take advantage of the fact that it will only be dealing with event sentences. Thus contextual clues that may have previously been inconclusive in nonevent sentences, are now more useful for identifying event roles.

One learned component in this model is the sentential event recognizer, which is a machine learning classifier relying on features in each sentence. Its decisions are based on various types of features, such as lexical items in the sentence, lexico-syntactic patterns, semantic classes, etc. The second learning component is the localized text extraction component, which identifies event roles within the event sentences. The localized text extraction component learns extraction patterns representing the local context around candidate extractions.

Because of the two independent components in the pipelined PIPER model, each component makes discrete decisions independent of the other. One problem that arises with this approach is that if the sentential event classifier incorrectly discards an event sentence, the localized text extraction component loses any chance of extracting event roles from these sentences. This all-or-nothing nature of the classifier can cause the model to miss out on such cases. This is especially bad for cases where strong local evidence could have overcome these errors from the sentential event classifier.

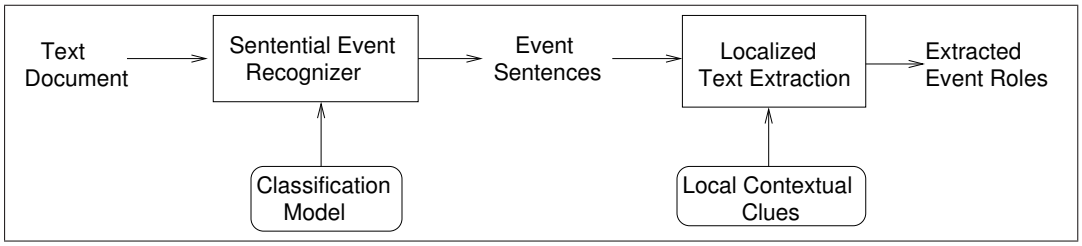


Figure 1.7: Block schematic of a two-stage pipelined model for IE (PIPER)

The second model for IE (GLACIER), presented in this research, seeks to overcome this drawback of PIPER by employing a probabilistic approach for combining evidence from the two sources. The GLACIER model also consists of two components, but it overcomes the limitations of PIPER, by unifying these two components in a probabilistic framework. Instead of making two separate discrete decisions about event sentences and event roles, this model computes two separate probabilities, which are combined by the unified model into a single joint probability. Decisions to extract event roles are based on this final joint probability. Figure 1.8 shows a block schematic of this model. By computing probabilities instead of discrete decisions, this model gently balances the influence of the two decisions.

The two learning components of this model are the sentential event recognizer and the plausible role-filler recognizer. The sentential event recognizer is again based on machine learning classifiers like in PIPER, the primary difference being that here it is used to generate probability estimates instead of a classification decision. The second learning component is the plausible role-filler recognizer, which analyzes local contexts of noun phrases and estimates the probability that they represent event roles of interest. This component also uses machine learning strategies for generating these estimates. Both components rely on various contextual features, such as lexical items, semantic classes, lexico-syntactic patterns, etc.

The remainder of this dissertation describes these approaches in detail, and empirically demonstrates the utility of event detection in IE. The two approaches presented in this research — a pipelined approach and a unified probabilistic approach — are shown to achieve this goal. Additionally, the dissertation goes on to show that using a probabilistic model to balance the influence of the two components overcomes the disadvantages of discrete and independent decisions made by the pipelined model.

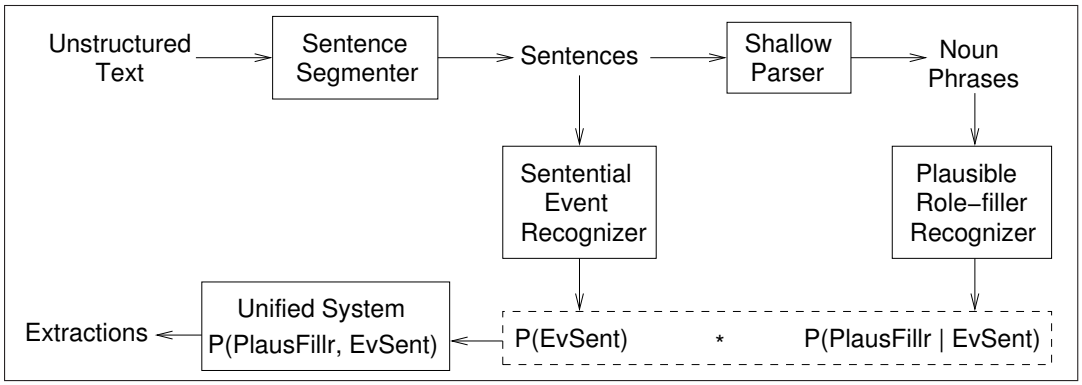


Figure 1.8: Block schematic of a unified probabilistic model for IE (GLACIER)

1.4 Claims and Contributions

The scientific contributions of this research to the field of event-based information extraction are as follows:

1. *An explicit sentential event recognition process can benefit IE.*

   The primary motivating factor underlying this dissertation is the observation that humans can effortlessly identify role players of events in text documents despite the lack of direct evidence of an event or an event role in the immediate local context of many of the candidate phrases. This research hypothesizes that an automated system could extract such role fillers by detecting event descriptions in wider regions of text (e.g., sentences or paragraphs). Given the knowledge of an event description within a region of text enables an IE technique to make more confident decisions about event role extractions in spite of weak local contextual evidence.

2. *Making joint decisions through a probabilistic model for IE improves performance over a pipelined model.*

   This dissertation presents an approach for IE that combines probabilistic information about event sentences with probabilistic information about role fillers into a joint probability, which is used for making extraction decisions. This approach enables the model to gently balance the influence of the two components to achieve better IE performance. By using a probabilistic approach, this model overcomes the drawbacks of a pipelined approach for combining the two components. The use of probabilities overcomes the discrete nature of the pipelined model, and the decisions on a joint probability overcomes the problems from the independent decisions of the pipelined model components. Overall we achieve better performance with the unified probabilistic model over the two-stage pipelined model.

In addition to these primary contributions, there are several ancillary contributions that result as a by-product of this research:

1. This work presents an analysis of event sentences in text. A human annotation study illustrates the complexity of the task, and provides insights into event descriptions in text. High agreements in this annotation study illustrates the feasibility of the task, and this study has resulted in annotated data sets of event sentences in two domains.

2. Several machine learning classifier-based approaches (supervised as well as weakly supervised) are shown to be suitable for identifying event sentences in text. Some

of these classifiers are also used to generate probabilistic estimates in the unified probabilistic model for IE.

3. This research investigates the contribution of several types of contextual clues in identifying event descriptions in text, as well as indicators of event roles within these descriptions.

4. This research investigates weakly supervised approaches for the components of our model — recognizing event sentences and learning extraction patterns.

1.5 Navigating this Dissertation

The following list summarizes the chapters in this dissertation to help guide the reader through this document:

- Chapter 2 describes existing work in IE and presents a discussion of how the research presented in this dissertation relates to previous work. In addition, this chapter presents information about IE tasks and data sets, to provide a background for the research presented in this dissertation.

- Chapter 3 presents a detailed description of the pipelined model for IE[3] called PIPER. The chapter describes the two components of the model, discusses the features used in the model, and presents supervised and weakly supervised approaches for training these components.

- Chapter 4 presents a detailed description of the unified probabilistic model for IE[4] called GLACIER. The chapter describes machine learning approaches for computing probabilities for the two components of the model, and shows how these probabilities can be combined for making better decisions about event role extractions.

- Chapter 5 discusses event sentences and their relationship to event roles. The chapter contains the description of a human annotation study for event sentences — the guidelines, the annotation interface, the agreement study, an analysis of the annotated data and a discussion of findings from this analysis. The sentential components of the two models are then evaluated in the last part of this chapter.

---

[3]We have previously described some of this work in a conference publication [82].

[4]We have previously described some of this work in a conference publication [83].

- Chapter 6 contains an empirical evaluation of the two IE models — PIPER and GLACIER. The two models are each evaluated on two domains and a comparison of their performance with other baseline approaches is presented here.

- Chapter 7 presents a summary of conclusions from this dissertation research, recounts the list of scientific contributions of the research, and finally discusses potential directions for future research.

CHAPTER 2

BACKGROUND

Many ideas have been proposed by researchers over the years to extract various forms of information from free text. This chapter briefly summarizes these techniques, which are compared and contrasted against the research presented in this dissertation. This chapter first presents a brief history of information extraction in Section 2.1, and the event role extraction task in Section 2.2. This is followed in Section 2.3 by a summary of the various approaches that have been applied to IE (and event role extraction). Some current approaches that attempt to incorporate global information into IE, and techniques for text classification or topic detection are then described in Section 2.4 and Section 2.5, respectively. Finally, Section 2.6 describes the two data sets for event-based IE that are used in this research.

2.1 A Brief History of IE

Information extraction, in one form or another, has been an important problem since the early days of NLP. It has stemmed, primarily, from the need for people to more easily organize and manage the vast amounts of information described in free text. Free text contains a multitude of information that, if effectively extracted, can be useful for many real-world applications. These useful pieces of information are of many different forms — name of people, places, organizations, roles played by entities in events, relations between entities, etc. These are characterized by the fact that they can be organized into structured formats, such as database entries.

Some of the early work in information extraction has been nicely summarized by Lehnert and Cowie [63], who mention the work by DeJong [30, 31] on analyzing news stories among the early attempts at IE. The system, called FRUMP, as described by DeJong, is a general purpose NLP system designed to analyze news stories and to generate summaries for users logged into the system. This system is strongly reminiscent of modern day IE, since the generated summaries are essentially event templates filled in by FRUMP and presented as single sentence summaries of the events. DeJong's system used hand-coded rules for

"prediction" and "substantiation" (the two components of his system) to identify role fillers of 48 different types of events. FRUMP uses a data structure called "sketchy script," a variation of "scripts" previously used to represent events or real-world situations described in text [100, 26].

Other approaches for information extraction in this era include a Prolog-based system by Silva and Dwiggins [103] for identifying information about satellite-flights from multiple text reports. Similarly, Cowie [24] also implements a system, based on Prolog, using "sketchy syntax" rules to extract information about plants. By segmenting the text into small chunks, according to pivotal points, like pronouns, conjunctions, punctuation marks, etc., the system avoids the need for complex grammars to parse text. Another system developed around the same time was that by Sager [98], which applied to highly domain-specific medical diagnostic texts (patient discharge summaries) to extract information into a database for later processing. The system uses English grammar rules to map the text into a structured layout. Also of note is the work by Zarri [128], whose goal was to identify information about relationships and meetings of French historical personalities and represent this information in a more structured form in the "RESEDA semantic metalanguage." The system uses rules for semantic parsing and heuristic rules of identifying slot-fillers required by the RESEDA metalanguage. Thus, a fair amount of effort was put into IE during this period of NLP research. Much of this work focused on specific domains, used hand-crafted rules and did not have standard data sets or standard evaluation procedures.

To encourage the development of IE techniques, in the late 1980s and early 1990s the US Government (DARPA) organized a series of Message Understanding Conferences (MUC) [46] as a competitive task with standard data and evaluation procedures. Running from 1987 through 1997, the first few MUCs defined a single IE task for the participating teams. In the later conferences, IE was separated into several different tasks. The various types of IE tasks defined there differ primarily in their degree of complexity and in the depth of information extracted. For instance, the *named entity* (NE) task is that of identifying within free text, person, location and organization names, and quantities, such as dates, monetary amounts, etc. The tasks then get more complicated with the *coreference task* (CO) that involves the identification of coreferent entities in text, and the *template elements* (TE) task of discovering specific attributes about these entities. Next, the *relation extraction* (RE) task requires the detection of specific relations (such as *employee of*, *author of*, etc.) between the discovered entities. Finally, the most complex of these IE tasks, is the *scenario template* (ST) task, which requires the system to identify instances of a specific predefined

event in the text, and extract information pertaining to each instance of the event found. The system is expected to output an event template containing various pieces of event information corresponding to each event detected within the given text. Thus, a wide range of intricacy exists in locating the various forms of interesting information embedded in free text.

Many systems (e.g., GE [60], SRI [49], UMass [62], NYU [45], etc.) participated in these shared tasks, which greatly benefitted IE research. These conferences established standard data sets for the development and comparative evaluation of systems designed to perform IE. They spurred great interest in this field of research, and led to the formal definition and evaluation criteria for various different kinds of tasks related to IE. By establishing a competitive environment, the MUCs enabled rapid transfer of ideas and techniques between teams, with an overall improvement in IE technology. A majority of the IE techniques developed during the MUC conferences were based on hand-coded rules and massive amounts of knowledge engineering by human experts. However, during the later years of MUC, efforts [91, 92] were also being focused on reducing the amount of human undertaking involved in generating rules for IE. Such research work [124, 110] has continued over the years and succeeded in drastically reducing the human involvement in building an IE system for a specific domain.

Another DARPA program introduced towards the end of the MUC era was the TIPSTER program [73, 2, 41] designed to advance the state of the art in text processing. IE technology benefitted from this because of the development of several standard IE architectures and toolkits (such as GATE[1] [28]) with the goal of having standardized and reusable components for use across large IE systems.

Research in IE has continued to grow over the years since MUC and TIPSTER. The definition of IE has also gradually broadened to include many different types of information and tasks that differ in their complexity, the amount of resources used, their training methodologies, etc. Researchers are looking at extracting information from semistructured texts, such as resumés [126] and classified advertisements [42]. Such texts also lend themselves to a "field segmentation" model for IE, which is vastly different from the "nugget extraction" task seen in event-based IE. Recent approaches to IE are also looking to incorporate machine learning techniques [39, 15, 37, 64, 36] for information extraction. Finally, recent trends in IE [69, 55] are looking to incorporate more global information into IE systems than was possible with the hand-crafted pattern-based approaches from the

[1] http://gate.ac.uk

MUC systems. From this wide range of IE tasks and approaches, this dissertation focuses on event-based IE, with the goal of using information from the wider context to make better inferences about roles played by entities in events.

2.2 Event Role Extraction

The research presented in this dissertation focuses on the scenario template (ST) task [114, 115] of extracting event role fillers for a specific type of event. The type of event is specified beforehand. This specification of events includes fine-grained details about the circumstances under which a textual discourse is said to contain an event description. In the case of disease outbreak events, for example, the specification of the event type should indicate the exact nature of a textual description that would constitute a disease outbreak. Perhaps, a single report of a common disease is not typically considered a "disease outbreak." Thus, the event-type specification must unambiguously exclude such cases. Given such an event specification, and some training examples of documents along with their corresponding event templates, the goal of the designed IE systems is to identify instances of the specified event-type in new texts and identify event role filler string within these event descriptions. Note that a single document can contain multiple instances of the specified event-type.

While the complete ST task is to generate "event templates" using the extracted role fillers, this dissertation focuses on the role filler extraction part of the process. As shown in Figure 2.1, for the complete ST task, the system must generate an event template corresponding to each event detected in the text. A typical solution to address this task is to first attempt to locate all spans of text within a document that constitute potential role fillers for the defined event-type. The extracted strings are then placed into one or more templates by a template generator, each template corresponding to one distinct event. Since the goal of this dissertation is to accurately identify event role filler strings in text, a template generation component is not applied to the extracted strings. The exact evaluation procedures employed in this research will be discussed later in the dissertation.

The complete ST task is challenging, and many different approaches have been explored by researchers, with varying levels of success. For identifying event role fillers, these approaches include pattern-based systems that attempt to match the contexts in which the role fillers occur, and machine learning systems that rely on features in the context of candidate role filler text spans. Some of these systems also use various amounts of knowledge engineering, to incorporate human knowledge into the system. Thus, current approaches

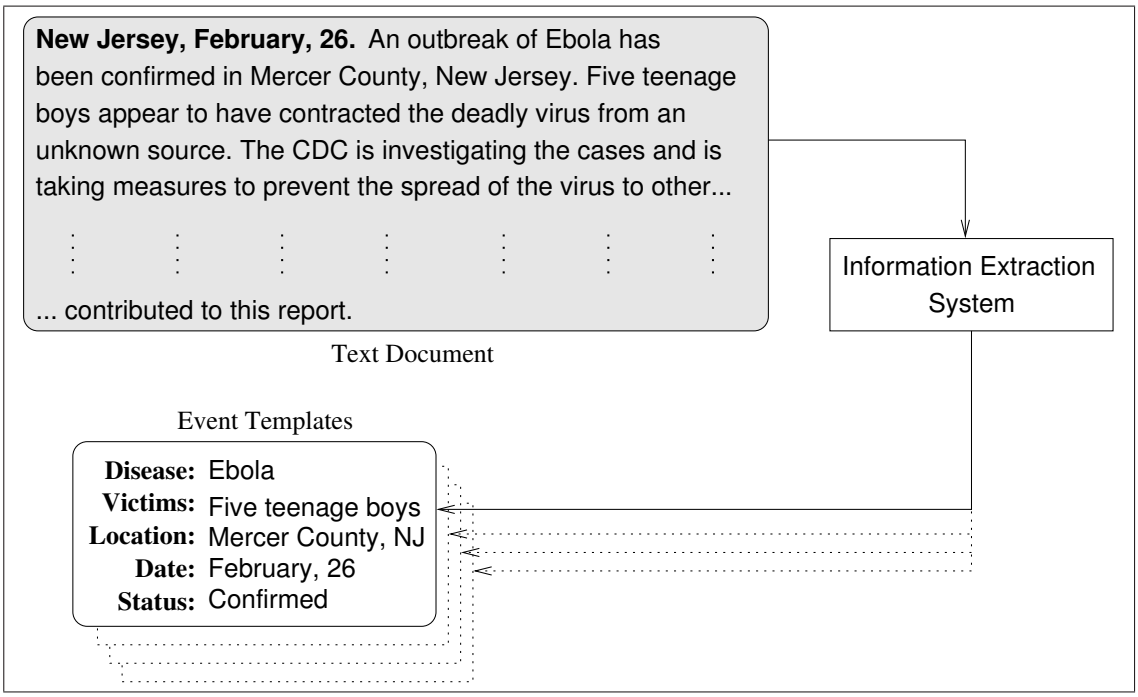


Figure 2.1: Block schematic of the scenario template (ST) task

for the ST task can be generally organized according to their approaches for text extraction and template generation. They can, alternatively, also be categorized by the amount and type of knowledge engineering required for the system, or by the types of training strategies used in the system.

Despite this clear description of the ST task, there is still a great variation in complexity even within this task. This typically happens due to the nature of the different types of events that may be defined for the task. In some cases the event information is embedded within semistructured text, which is easier to extract than pieces of information that are sparsely distributed within free text. For example, extracting information from resumés is fundamentally different from extracting information about disease outbreaks from news articles. In the case of resumés, the text is somewhat structured, and the information is more densely distributed within the document. Further, a resumé typically contains information about a single person — a fact that can be exploited in the extraction task. On the other hand, for disease outbreak events, the relevant information is sparsely distributed, there is no predefined structure to the texts, and multiple disease outbreaks can be described within a single document. All of this makes IE for such cases vastly more difficult. This dissertation attempts to address this more complex case of identifying event information in free, unstructured text.

Another distinction between the various approaches for IE is the nature of the training and the amount of knowledge engineering required. Many of the systems designed for IE tasks require some amount of knowledge engineering in the form for creating semantic class dictionaries, or extraction pattern lists. Pattern-based algorithms for this problem try to automatically or semiautomatically learn extraction patterns for locating relevant information. Similarly, classifier-based methods used machine learning models to extract spans of text. All of these methods rely on varying quantities and forms of training data. Thus, the current state of the systems for IE includes numerous variations and possibilities, which will be discussed here.

2.3 Overview of IE Techniques

Owing considerably to the MUC conferences, a fair amount of effort has gone into IE research in recent years. Most of the earlier approaches to the problem were pattern-based approaches, which use contextual patterns to extract relevant information from text [57, 91, 106, 52, 50, 92]. These contextual patterns attempt to match lexical, syntactic and semantic characteristics in the vicinity of a potential extraction. Recently, the trend has been to employ machine learning techniques for this task. One way of applying standard machine learning approaches is to use them to learn extraction patterns [17, 97, 12, 71]. Another is to tackle the IE problem as a "sequence tagging" problem and use existing sequence labeling or classification systems for this [39, 15, 37, 64].

Despite this large variation in the approaches to IE, it appears that the strategies are highly correlated to the nature of the data. Machine learning techniques seem to work well for texts that are somewhat structured or semistructured, such as resumés, classifieds, bibliographies, etc. Because semistructured documents usually contain more fragments of text that cannot be parsed, techniques that extract information from these usually do not do a linguistic analysis of these texts and usually tend to use sequence tagging approaches. On the other hand, for unstructured free text, where the information to be extracted is very sparsely distributed within the text, predominantly pattern-based approaches have been successful. The typical examples of these are everyday event descriptions in newspaper articles, books, etc. Systems that deal with free text (newspapers, books, etc.) usually tend to do linguistic analysis for IE.

2.3.1 Pattern-based Approaches

Typically, pattern-based strategies [124, 87, 111, 116, 110] employ patterns that rely on linguistic analysis of text. These methods differ primarily in two respects:

(a) The expressiveness of their patterns.

(b) The learning technique used to generate patterns.

The expressiveness of the patterns is tied to the linguistic analysis done by the system and to the knowledge engineering involved. The patterns could be hand-created or they could be learned automatically or semiautomatically.

Earlier systems (such as PALKA [57], FASTUS [4], AutoSlog/AutoSlog-TS [91, 92], CRYSTAL [106], LIEP [52]) inspired by the MUC conferences essentially laid the foundation for pattern-based IE. The PALKA system [57] composes a set of extraction patterns for an IE task using sentence-level and clause-level relevance judgments (by a human expert) of training text. Relevant sentences along with semantic class assignments to words are used to generate IE patterns. This is somewhat related to our research, in that the PALKA system uses "relevant regions" (sentences/clauses) to learn IE patterns. However, finding the relevant regions requires some human oversight, and are only used in training, and not in the end system. The FASTUS system [4] uses hand-made extraction patterns encoded into a finite state transducer for IE. CRYSTAL [106] uses training examples to learn its set of extraction patterns (or "Concept Node definitions"), which are then generalized using similar patterns from the text. Similarly, AutoSlog [91] relies on the answers in the answer key templates to generate extraction patterns for a given task. All of these systems use patterns representing a tight local context surrounding the desired extractions.

The AutoSlog-TS [92], which is employed in this research as a baseline and for pattern generation, creates lexico-syntactic patterns based on a shallow parse of the text. The patterns are designed to match constituents in the text, and one or more of the matched constituents is extracted. Its pattern learning process is minimally supervised, in that it requires only a set of relevant and irrelevant documents for learning. The learning process relies on the distribution of patterns between the relevant and irrelevant documents to generate a ranking for the patterns. The AutoSlog-TS system is a successor to the AutoSlog system [91], which did not have the benefit of a pattern ranking and thus required more human effort for pattern selection. The primary drawback of both systems (AutoSlog and AutoSlog-TS) is the requirement of a human expert to inspect patterns, to assign patterns to specific event roles, and to discard the inferior patterns.

Many recent systems [124, 87, 111, 110] are attempting to use minimal supervision, in the form of manually created seeds or queries, for pattern learning. The patterns used by Yangarber et al. [124] rely on a partial parse of the text, which is enhanced with some domain-specific semantic information. The primary contribution of this work is the

pattern discovery process, which uses only a set of seed patterns and a set of domain-specific documents for the learning process. Stevenson and Greenwood [110] use pattern structures similar to those by Yangarber et al. They also use a set of seed patterns to initiate the pattern induction process. However, rather than using the relevance of the documents to rank new patterns, they define a pattern similarity measure based on word and concept similarity measures. From the set of patterns generated from a document set, they iteratively select the most similar patterns to the seed patterns and add those to the seed set.

Sudo, Sekine and Grishman [111] introduce a new pattern model for IE, and show that it performs better than two existing models. They use queries to an IR system to collect documents that are relevant to the task. Patterns are generated from these using dependency parse links between named entities, and are then ranked based on a TF-IDF function and filtered on frequency cutoffs. Poibeau and Dutoit [87], on the other hand, use patterns based on predicate-argument structures. Their system also uses a set of seeds to locate new patterns in unannotated text. The pattern discovery process uses the seed patterns to look for other similar patterns in the text by locating words in identical syntactic constructs. This is done by using a semantic network, and by expanding upon the existing pattern set using syntactic variants.

Similarly, there exist IE systems that require greater supervision for pattern discovery. Surdeanu et al. [116] employ a very unique approach to pattern discovery using a machine learning approach. Their extraction patterns are also based on predicate-argument structures found in a full parse of the text. They train a decision tree algorithm on fully parsed and annotated training data to automatically identify argument constituents of the predicate, and to automatically map the arguments to roles. Constituents corresponding to specified roles are then extracted.

Many systems [17, 97, 12, 71] use similar learning strategies to learn rules or patterns from annotated data. An example of a rule used by Califf and Mooney [12] is shown in Figure 2.2. All of the rule-based systems do some word-level processing of text, such as part-of-speech tagging, morphological analysis, and semantic tagging of words. Ciravegna's system [17] learns separate rules for finding the start of extractions and the end of extrac-

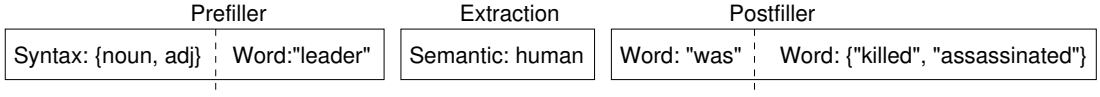


Figure 2.2: An example of a pattern matching rule for the RAPIER system

tions, and uses a covering algorithm to learn rules from an annotated training set. The system by Roth and Yih [97], and by Califf and Mooney [12], both use some variation on relational learning, to learn extraction rules from training texts.

2.3.2 Sequence Tagging Approaches

Sequence tagging approaches view the documents as a sequence of tokens and the IE task is formulated as identifying subsequences of the tokens as extractions. This is typically done by training a Machine Learning classifier on training data to classify each token as belonging to an extraction or not. These approaches differ primarily in the Machine Learning approach that is used, and the features used for training. These approaches work particularly well for semistructured texts where the information can be found in "fields," and where parsing is difficult.

A system by Freitag and McCallum [39] uses Hidden Markov Models (HMMs) for identifying the extraction strings in the documents. The probabilities for the HMM are gathered from the training data. Some of the states of the HMM are categorized as target states and some are not. During the classification phase, it determines the sequence of states that has the highest probability for generating the document. The tokens which are associated with the target states are deemed to be the extractions. The most significant contribution of this work is that the system automatically builds the structure of the HMM by stochastic optimization using the training data. Following the success of this work, other systems [126, 47] implemented some variations based on HMMs. Yu et al. [126] introduced a cascaded multilevel approach to process resumé data. Gu and Cercone [47] described a segment-based approach to improve performance.

Chieu, Ng and Lee [15] compare three different Machine Learning systems used for IE. More importantly, they do a deeper linguistic analysis of text using a parser, and include a number of features based on the parse trees as input to the learning algorithms. Their algorithm operates on a "sequence" of syntactic phrases or chunks, unlike most other sequence tagging systems, which process streams of words or tokens. Chieu, Ng and Lee show that using these syntactic features improves performance.

Many of the sequence tagging approaches employ SVMs [37, 64] for IE because of their proven track record. The algorithm by Finn and Kushmerick [37] applies two stages of classification. The first stage uses a high precision SVM classifier to tag the start and end boundaries of each extraction. The second stage uses a high recall SVM classifier to then identify extractions for cases where one of the boundaries may have been missed by the first stage classifier. This two-stage approach improves the coverage and the overall performance

of the IE system. Li et al. [64] attempt to overcome the imbalance between the number of positive and negative training examples for IE by experimenting with *uneven margin* SVM and Perceptron. Both of these approaches use word-based features such as tokens in the local context, parts-of-speech, semantic tags and surface features like capitalization.

Recently, researchers have also used CRFs [36, 104] for IE. These approaches primarily rely on lexical features like word-types, parts-of-speech and semantic tags. Finkel, Grenager and Manning [36] attempt to capture long distance relations in language by using nonlocal features obtained through *Gibbs Sampling*. Bunescu and Mooney [8] use a variation of CRFs called Relational Markov Networks to exploit relations between features for learning.

2.3.3 Other Related Approaches

While a great deal of research has gone into identifying role fillers of events from semistructured and unstructured texts, there is also a substantial amount of related work in other areas of NLP that should be acknowledged.

One variation of event-based IE is the task of "relation extraction" which is the task of identifying specific relations between entities mentioned in text. It is different from event-based IE, because the relations to be extracted are not tied to specific events. Rather these relations are real-world "facts" described within text documents. For example, the location of a company is a factual relation between that company and the city (or state). Maslennikov, Goh and Chua [70] use a pattern-based approach using dependency relations for this task. Similarly, Zhou and Zhang [130] take a machine learning approach, applying SVMs for relation extraction.

Most the IE work described so far has been targeted towards predefined tasks, relations or event types. However, in a variation of traditional IE, called Open Domain Information Extraction, the set of relations are not defined beforehand. Rather it is the job of the IE system to discover the set of relation types, before extracting instances of these relations between entities from free text. Systems designed for this task rely on redundancy of relations in large corpora in identifying and extracting relations. For instance, the ODIE system [101, 102] exploits redundancies in a corpus (typical of the Web) to extract relations pertaining to a user query. Similarly, with the growth of the Internet, a number of systems attempt to exploit the vast quantity of free text available on the Web for IE. The KnowItAll system [34] uses the Web to learn named entities and relations between them with the goal of automatically creating an ontology. Large text resources such at the Web, however, need not be limited to relation extraction. Patwardhan and Riloff [81] show

how the Web can be used to learn event-based extraction patterns, which could augment an existing IE system.

2.4 Incorporating Global Information

One common observation about the systems designed for IE is that most of them focus only on the local context surrounding the candidate strings that are under consideration for extraction (see Figure 1.2 and Figure 2.2 for examples). However, many researchers are realizing the importance of global information for this task. Focusing on the local context can lead to some ambiguity regarding the relevance of the text. Also many discourse cues can provide additional information about the relevant events. Thus, some recent approaches have made attempts to incorporate global information.

Recent work by Maslennikov and Chua [69] describes a technique for including long distance relations into pattern-based IE by using discourse cues. They use an existing discourse parser [107] to generate discourse links between different parts of each sentence, and between sentences. Such long distance discourse links are then used to connect local dependency relations from different parts of the document. This leads to improved performance for IE. However, their work focuses on a specific types of global information, while the research presented relies on event recognition in sentences as a source of global information for IE.

Other attempts at including global information into IE have also been made. Finkel, Grenager and Manning [36] use a sequence labeling model for IE. Their research incorporates nonlocal information into their model by enforcing label consistency within a document using Gibbs Sampling. Their hypothesis is that within a document different mentions of the same entity should get the same label. Similarly, Chieu, Ng and Lee [15] use a machine learning system with a wide range of features to locate the desired information. Coreference links between entities in the text is one of the features used by them to take advantage of the global discourse cues provided by coreference. Again, unlike the research described here, these systems rely on very specific types of global information for IE.

Finally, a related strategy for IE employed by Xiao, Chua and Cui [123], while not actually attempting to incorporate global information into the IE system, does try to address one of the problems of current IE systems mentioned in Chapter 1. Since most extraction patterns are not completely accurate in all contexts, their research describes a technique of selectively applying extraction patterns using, what they call, soft matching patterns. Soft matching patterns are probabilistic extraction patterns learned using a weakly supervised bootstrapping algorithm. The research presented here also plans to

selectively apply local contextual clues for IE, but our approach instead relies on the recognition of event descriptions in a wider context.

2.5 Relevant Regions in Text

The IE system presented in this research intends to incorporate global relevance information into an IE model by using an event sentence identification module before text extraction. Thus, to gather some insights about various possible techniques for sentence classification, this section will provide some background on this subtask. Specifically, we will look at different ways in which regions of interest within documents are identified in applications.

The problem of relevant region identification is essentially the problem of text classification at the sentence or region level. Thus, we could consider using standard text classification systems for this. However, most text classification systems are designed to work on entire documents, and not on smaller units of text, like sentences or paragraphs. These regions contain significantly less information compared to documents. Thus, classification of regions would require more specialized classification strategies.

To perform classification of regions of text without using annotated data, a class of machine learning systems called *multiple instance learning* [32] has evolved over recent years. These were originally developed for classifying segments of images without requiring segment-level annotations of images. The systems are trained using positive and negative *bags*. A positive bag is a set of instances of which at least one is positive, while in a negative bag all instances are negative. Systems train over the bags and then are able to assign instance-level labels during testing. Thus, multiple instance learning allows us to do without instance labels (or region-level annotation in our case). This greatly reduces the annotation cost involved. Recently, this type of learning has been used in natural language applications, such as text categorization [3] and relation extraction [11]. We employ one such strategy in our IE models for event recognition in sentences.

While the use of a text classification as the basis for recognizing event descriptions is investigated in our IE models presented later in this dissertation, there exist a number of other related approaches for identifying relevant sentences, such as information retrieval and text summarization techniques, which need to be acknowledged here.

In Information Retrieval (IR), the primary focus of researchers is to effectively and accurately retrieve information that is relevant to given queries from large document collections. The most common examples of this are the commercial search engines used by millions of people to locate relevant pages on the Web. IR systems have also been applied in natural

language tasks, such as Question Answering (QA) systems. In such systems, an IR module is used to retrieve passages that are relevant to a given question. An answer to the question can then be extracted from these retrieved passages. Thus the QA systems, in essence, contain a relevant region identification phase as part of their strategy.

Passage retrieval with respect to an input query is a well-studied problem in IR. Techniques to address this problem attempt to match relevance of passages within a collection of documents with the text queries. Commonly, the matches between queries and passages are weighted by the *IDF* values associated with the matching words [19]. Such techniques for passage retrieval have been extended by using other measures of "matching" to improve retrieval. For example, a research group at IBM [53] uses synonyms from a thesaurus and a number of measures based on the distribution of words to determine relevance of passages. Tellex et al. [117] provide an in-depth analysis of a number of passage retrieval systems in the context of QA. All of these systems based on word matching are termed as "density-based passage retrieval." Likewise, researchers have also seen success by exploiting dependency relations between the entities [25, 112], language modeling [67] and HMMs [77] to perform passage retrieval.

Another natural language application that is closely related to region classification is that of *automatic text summarization*. One of the techniques for automatically generating summaries for text documents is to identify a subset of the most important sentences within these documents and output them as the summary of the document. This is closely related to our goal of identifying event sentences, with the primary difference being that the summarization systems are not built for specific events.

A minimally supervised approach for automatic text summarization is a Bayesian query-focused summarization technique [29], which uses the relevance between queries (in the context of Information Retrieval) and documents to define a measure of the importance of sentences, and a Bayesian model trained for identifying the important sentences. Similarly, researchers have also tried strategies that use no supervision [6, 5, 78, 74]. The idea behind these approaches is to use different kinds of "links" between sentences to define their strength or importance, and then extract the most important sentences as summaries. Azzam et al. [5], and Barzilay and Elhadad [6] use coreference chains and lexical chains to link sentences in text. Inherent characteristics of the chains and the text are used to define a measure of the strength of these chains, which can then be used to extract the strongest chain of sentences as the summary. Mihalcea [74] uses a graph-based approach for summarization, where sentences are the nodes of a graph and the edges are similarity values connecting

sentences. Sentences and nodes from this graph are then extracted for the summary using graph-based ranking algorithms used to assign weights to the nodes.

To sum up, we see a number of natural language applications that are indirectly tied with our goal of event sentence identification. The research presented in this dissertation, however, only investigates approaches based on text classification for event recognition in sentences. The other related approaches described here can open up several avenues for future research in this field.

2.6 Event-based IE Data Sets

While the methods and techniques developed in this dissertation are generally applicable to different types of events, here the models will be trained and evaluated on two domains to demonstrate performance and consistency of the methods across domains. Therefore, as this dissertation develops the models for IE, it will refer to examples and present discussions for these two domains. Consequently, the two domains — *terrorist events* and *disease outbreaks* — and the data sets for each are described here to provide better context for the ensuing discussions in this dissertation.

2.6.1 Latin American Terrorist Events

The *terrorist events* data set consists of documents about Latin American terrorist events, along with corresponding answer key templates. The data was created during the MUC-3 and MUC-4 conferences [113, 114] as a standard testbed for IE evaluation. A set of guidelines establish the notion of "terrorism," which includes several types of terrorist events, including *attacks*, *bombings*, *arson*, *kidnappings*, *robberies* and *forced work stoppages*. The MUC-4 guidelines define a terrorist event as:

> Relevant incidents are, in general, violent acts perpetrated with political aims and a motive of intimidation. These are acts of terrorism.

These include terrorist acts at various stages of accomplishment — *threatened*, *attempted* and *accomplished*. The data set was constructed from newswire articles or documents. A document may contain no relevant event description, in which case it is considered an irrelevant document and has an empty answer key template associated with it. Alternatively, a document may contain one or more relevant event descriptions, in which case it it considered a relevant document and has an associated event template for each event description contained within it. About 54% of the documents in this data set are relevant, while 46% are irrelevant.

Figure 2.3 shows a sample document and its corresponding answer key event template from the MUC-4 terrorist events data set. The document describes a terrorist attack on a Peruvian Defense Minister. The answer key template contains the key pieces of information of this event in slot-fillers. The official MUC-4 guidelines define about 24 different slots to be filled by role fillers for each event. Some of these slots are "string" slots that are filled by text strings taken verbatim from the document. For example, the perpetrators of the terrorist attack is filled by the string *THREE YOUNG INDIVIDUALS* taken directly from the document. The other type of slots are the "set fill" slots that are filled by entries from a fix set on possible values. For example, the type of terrorist event, selected as one of six possible values — *attack*, *bombing*, *arson*, *kidnapping*, *robbery* or *forced work stoppage*. As

DEV-MUC3-0011 (NOSC)

LIMA, 9 JAN 90 (EFE) — [TEXT] AUTHORITIES HAVE REPORTED THAT FORMER PERUVIAN DEFENSE MINISTER GENERAL ENRIQUE LOPEZ ALBUJAR DIED TODAY IN LIMA AS A CONSEQUENCE OF A TERRORIST ATTACK.

LOPEZ ALBUJAR, FORMER ARMY COMMANDER GENERAL AND DEFENSE MINISTER UNTIL MAY 1989, WAS RIDDLED WITH BULLETS BY THREE YOUNG INDIVIDUALS AS HE WAS GETTING OUT OF HIS CAR IN AN OPEN PARKING LOT IN A COMMERCIAL CENTER IN THE RESIDENTIAL NEIGHBORHOOD OF SAN ISIDRO.

LOPEZ ALBUJAR, 63, WAS DRIVING HIS OWN CAR WITHOUT AN ESCORT. HE WAS SHOT EIGHT TIMES IN THE CHEST. THE FORMER MINISTER WAS RUSHED TO THE AIR FORCE HOSPITAL WHERE HE DIED.

0.	MESSAGE: ID	DEV-MUC3-0011 (NCCOSC)
1.	MESSAGE: TEMPLATE	1
2.	INCIDENT: DATE	09 JAN 90
3.	INCIDENT: LOCATION	PERU: LIMA (CITY): SAN ISIDRO (NEIGHBORHOOD)
4.	INCIDENT: TYPE	ATTACK
5.	INCIDENT: STAGE OF EXECUTION	ACCOMPLISHED
6.	INCIDENT: INSTRUMENT ID	-
7.	INCIDENT: INSTRUMENT TYPE	GUN: "-"
8.	PERP: INCIDENT CATEGORY	-
9.	PERP: INDIVIDUAL ID	"THREE YOUNG INDIVIDUALS"
10.	PERP: ORGANIZATION ID	-
11.	PERP: ORGANIZATION CONFIDENCE	-
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	"ENRIQUE LOPEZ ALBUJAR"
19.	HUM TGT: DESCRIPTION	"FORMER ARMY COMMANDER GENERAL AND DEFENSE MINISTER": "ENRIQUE LOPEZ ALBUJAR"
20.	HUM TGT: TYPE	FORMER GOVERNMENT OFFICIAL / FORMER ACTIVE MILITARY: "ENRIQUE LOPEZ ALBUJAR"
21.	HUM TGT: NUMBER	1: "ENRIQUE LOPEZ ALBUJAR"
22.	HUM TGT: FOREIGN NATION	-
23.	HUM TGT: EFFECT OF INCIDENT	DEATH: "ENRIQUE LOPEZ ALBUJAR"
24.	HUM TGT: TOTAL NUMBER	-

Figure 2.3: Sample MUC-4 terrorist event document and template

seen in the figure, this slot is filled in by the value *ATTACK*.

Note that the event templates associated with the documents in MUC-4 are quite large and complex. Additionally, the set fill slots require techniques to map information from text to the set of possible values. The focus of the research presented in this dissertation is to identify event role slot fillers within a text document. Thus, we consider only on the string slots associated with the MUC-4 documents. Specifically, this research focuses on the following five string slots associated with terrorist events: *victim*, *human target*, *perpetrator individual*, *perpetrator organization*, *weapon*.

The MUC-4 data set consists of 1700 documents, divided into 1300 development (DEV) texts, and four test sets of 100 texts each (TST1, TST2, TST3, and TST4). We used 1300 texts (DEV) as our training set, 200 texts (TST1+TST2) for tuning, and 200 texts (TST3+TST4) as a test set. All 1700 documents have corresponding answer key templates. We will see later in the dissertation how these documents are used for the training and evaluation of the IE models presented in this research.

2.6.2 Disease Outbreak Events

The *disease outbreaks* data set [86, 82] consists of disease outbreak reports obtained from ProMed-mail,[2] an online reporting system for outbreaks of infectious diseases. For each of these documents, human annotators have created corresponding answer key templates containing various pieces of information pertaining to the outbreaks. In creating this data set, a list of guidelines was specified for the human annotators, defining the notion of an "outbreak." According to the guidelines:

> A template should be created only for reports of a specific outbreak. General descriptions of outbreaks (e.g., "a widespread outbreak of anthrax would cripple society") are NOT relevant.
>
> Only current/recent outbreaks are of interest, especially those ongoing at the time the article was written. Any outbreak described as having happened more than a year before the article is written should be avoided. There is a bit of leeway — for instance, if an article was written in July and refers to "last summer," that is close enough, and similarly 367 days ago is close enough. Use your judgment on these borderline cases.

[2] http//www.promedmail.org

> Outbreaks do not need to have identifiable victims. Outbreaks do need implicit or explicit victims, or have a confirmed organism or disease that creates a potential health hazard. The latter would include terrorist actions, such as anthrax mailings, even if no one was infected.
>
> Multiple outbreaks of the same disease in the same country, if authorities claim they are unrelated, should be marked up as separate outbreaks.

In this data, a report may contain no relevant event description, in which case it is considered an irrelevant document and does not have an answer key template associated with it. Alternatively, a document may contain one or more relevant event descriptions, in which case it it considered a relevant document and has an associated event template for each event description contained within it. About 82% of the documents in this data set are relevant, while 18% are irrelevant.

Figure 2.4 shows a sample document and its corresponding answer key event template from the ProMed disease outbreaks data set. The document describes an outbreak of Legionnaires' disease in Australia. The answer key template contains the key pieces of information of this event in slot-fillers. As was the case in the MUC-4 documents, here too some of these slots are "string" slots and some are "set fill" slots. For example, the disease

Another four people in Australia were Thursday confirmed with legionnaires' disease sourced to the Melbourne Aquarium, bringing the number of such cases to 91. So far the aquarium outbreak has claimed two lives while 19 people are still in hospital, six of them in critical condition. A 77-year-old man died Wednesday night following complications from the disease but he had not been confirmed as a victim of the aquarium outbreak, health officials said. The officials said they were unable to speak to the man and his relatives had not been sure whether he visited the aquarium during the 11-25 Apr 2000 danger period. Federal Finance Minister John Fahey is among those diagnosed with the illness and is recovering at home after visiting the aquarium for a Liberal Party function last month. The outbreak is the largest [outbreak of legionellosis] in Australia, although the worst [outbreak] was in 1987 when there were fewer cases but 10 people died.

Story:	20000514.0753
ID:	1
Date:	June 14, 2000
Event:	outbreak
Status:	confirmed
Containment:	—
Country:	AUSTRALIA
Disease:	legionnaires' disease / legionellosis
Victims:	91

Bytespans (Template 1): 444-466 1229-1242 540-542

Figure 2.4: Sample ProMed disease outbreak document and template

slot is a string slot filled by the disease string *legionnaires' disease* taken directly from the document. In contrast, the status slot is a set fill slot, which can be filled with one of *confirmed* or *not confirmed*. As seen in the figure, this slot is filled in by the value *confirmed*. Also note that these templates contain byte-span references to the exact position in the text where the strings can be found. These byte-spans are not utilized in this research. Again, following the arguments presented for the MUC-4 data, here as well this research focuses only on the string slots: *disease* and *victim*.[3]

The disease outbreaks data set consists of 265 ProMed articles with corresponding answer key templates. Of these, we use 125 as a training set, 20 as a tuning set, and 120 as the test set. Most of the ProMed articles contain email headers, footers, citations, and other snippets of non-narrative text, so a "zoner" program[4] was used to automatically strip off some of this extraneous information. All of these documents (relevant and not relevant) and the answer keys comprise the second data set that is used later in this dissertation to demonstrate the prowess of our IE models that were developed as part of this research.

---

[3]The "victims" can be people, animals, or plants that are affected by a disease.

[4]The term *zoner* was introduced by Yangarber et al. [125].

CHAPTER 3

PIPELINING EVENT DETECTION AND EXTRACTION

The goal of this research is to automatically identify event roles in text, especially in cases having weak local contextual evidence of their role in an event. The hypothesis is that knowing about event descriptions in regions of text can enable the IE model to make inferences about event roles even with weak or inconclusive evidence of the role in the local context. Chapter 1 presented a brief overview of PIPER, a two-stage pipelined model that realizes this idea. This chapter presents a detailed description of this model. Section 3.1 presents the outline of the pipelined approach, and discusses the characteristics expected of the two primary components of the pipeline. Section 3.2 and Section 3.3 cover the details of the *sentential event recognizer* component of the PIPER model, including training strategies and features used. Section 3.4 then emphasizes the purpose of weak local evidence in event sentences, and describes pattern-based methods for the *localized text extraction* component of PIPER. Section 3.5 presents a variation of the pipelined approach, where selective application of patterns by the localized component can further improve IE performance. Finally, Section 3.6 puts these components together into a single pipeline, and discusses the characteristics of various configurations of the pipeline model.

3.1 Event Detector and Extractor Pipeline

Most current IE systems rely on direct evidence of event role fillers in the context surrounding the candidate phrases in text documents. For example, a verb phrase like *was assassinated* provides evidence of a criminal act described in the document, and the subject of this verb phrase can be quite reliably extracted as the victim of that criminal act. However, a many event role fillers have no direct evidence in their local contexts linking them to the described event. In a majority of these cases, human readers make leaps of inferences about role fillers based on indirect evidence from the wider context. The goal of this research is to automatically identify such role fillers for IE.

Many of the inferences made by human readers are based on the recognition of event

descriptions in the wider context. Take the text snippet[1] in Figure 3.1 for instance, which describes the scene in the aftermath of Typhoon Morakot in China. Observe that the news story first describes the state of the human casualties (*people feared dead*), and the property damage (*buildings have toppled over*), and then goes on to say that a typhoon hit parts of Taiwan and China. These two pieces of information are loosely connected by the preposition "after," and human readers have no trouble making the causal connection between the typhoon and its destructive effect on humans and property.

For an automated IE system that relies solely on local contextual evidence, however, making such inferences in nontrivial. Using the local context surrounding the candidate role fillers for this event, an automated IE system can determine the state of the entities mentioned (people who are *feared dead* or *stranded*, and buildings that are *toppled*). To determine that these entities are role fillers in a particular event, the system needs to infer that the cause of their state is the event of interest. This information appears later in the sentence, and must be used to make the correct inference about event roles of the event.

This research presents an approach for automatically making such inferences based on the recognition of event descriptions in the wider context. The approach comprises of a *sentential event recognizer*, whose job is to identify sentences containing descriptions of events of interest. This information can then be used by the *localized text extraction* component to make inferences about the roles played by entities within these sentences. Thus, in the example presented in Figure 3.1, our PIPER model for IE first determines that the sentence is describing a natural disaster event. Then, the localized text extraction component makes the inferences that the people who are feared dead are the victims of this natural disaster, and the toppled buildings are the property damaged in this event. The two components of this IE model are organized in a pipeline, such that the output of one component is "piped" as input to the second component.

The inspiration for a pipeline approach implemented in the PIPER model for IE comes from various NLP systems that employ a cascaded strategy for language processing. Putting

> Hundreds of people are feared dead, thousands stranded, buildings have toppled over and more than a million people have fled their homes after Typhoon Morakot lashed Taiwan and China with heavy rains and strong winds.
>
> In China, the storm triggered a massive landslide in eastern Zhejiang province that toppled six apartment buildings and buried an unknown number of residents.

Figure 3.1: Text snippet illustrating inferences made by human readers

[1]Quoted from *Al Jazeera* online English edition (http://english.aljazeera.net), August 11, 2009.

components in a pipeline, with the output of one processed as the input by the other is a natural way to organize the modules. This is a common practice in many NLP applications, where data are sequentially processed by cascaded data processing engines. In addition, this organization can also be seen as a filtering operation, which enables the pipelined extraction model to focus on the text of interest. A similar filtering paradigm is presented by Riloff et al. [96], who preprocess the input text with a sentiment analysis system to improve the performance of pattern-based IE. Similarly, typical Question Answering systems [89, 20, 48] employ a pipeline approach, consisting of a document or passage retrieval stage followed by an answer selection or answer extraction stage. Following these examples of pipelined systems, this research implements the sentential component and the local component in a pipelined or cascaded framework for IE.

The basic idea of this research is to detect event descriptions in regions of text, and use this information to make better extraction decisions. There are many possible definitions for *region of text* (e.g., Salton et al. [99], Callan [13]), and exploring the range of possibilities is an interesting avenue for future work. This research simply uses sentence boundaries to define regions in text. This has the advantage of being an easy boundary line to draw (i.e., it is relatively easy to identify sentence boundaries) and it is a small region size yet includes more context than most current IE systems do.[2] With this definition of regions, the specific aims of this research are to identify sentences in text that discuss events of interest, and then use local context to extract event roles from these sentences.

The PIPER model for IE consists of two components to achieve these objectives. The components operate on the input text independently of one another, which has some benefits for the overall system. For one, decoupling these tasks simplifies the learning process. Identifying event sentences amounts to a text classification task, albeit the goal is to identify not just relevant documents, but relevant subregions of documents. Within an event sentence the local context may not need to be as discriminating. So a more general learning approach may suffice for the local contextual component. The two components of the PIPER IE model are organized in a pipelined framework, where one component processes the input text and sends its output to the second component for further processing.

Figure 3.2 presents an overview of PIPER, the two-stage pipelined model for IE. First in the pipeline, the *sentential event recognizer* operates on input text documents to detect event sentences. The *localized text extraction* module that follows in the pipeline extracts event roles, and can now capitalize on the fact that it will only be dealing with the event

[2]Most IE systems consider a context window of a few words on either side of a potential extraction.

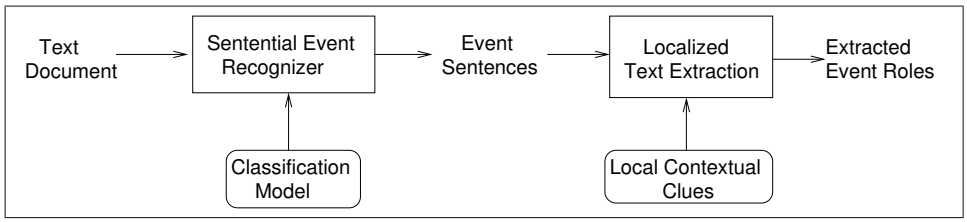


Figure 3.2: Overview of PIPER — a pipelined model for IE

sentences in the input text, and not with the entire text document itself. Thus, contextual clues that may have previously been inconclusive in nonevent sentences, can now be more useful for the localized component in identifying event roles.

The task of the first component in the pipeline, the *sentential event recognizer*, is very similar to that of text classification systems, with the primary difference being that here we are dealing with the classification of sentences, as opposed to entire documents. Following previous work in text classification, this research takes a machine learning approach for the classification decisions of the sentential event recognizer. The module assesses the strength of various features in each sentence and decides on a class for the sentence — *event sentence* or *nonevent sentence*.

The task of the *localized text extraction* component of the IE model is to extract event roles from the event sentences identified by the sentential event recognizer. This is similar to the task performed by current IE systems. However, one main difference here is that the localized text extraction component in this IE model is applied only to event sentences identified by the first component. This enables the localized component to use weaker evidence in making decisions about event roles, based on the knowledge of event sentences from the sentential event recognizer. The localized text extraction component of the PIPER model analyzes various local contextual features associated with a word or a phrase to decide its role in an event, *accounting for the fact that the word or phrase appears within an event sentence*.

Thus, the goals of this pipelined approach for IE are to train the two components such that they are applied independently of one another, but the localized text extraction component can benefit from the decisions made by the sentential component. The remainder of this chapter explores strategies for training these two components, addressing issues of contextual representation, features, learning algorithm, training data, extent of supervision, etc. The two trained components are then put together to make up the PIPER model.

3.2 Sentential Event Recognizer

The sentential event recognizer is tasked with identifying event sentences in text documents. Given a new (unseen) text document, a sentence boundary detector segments the document into individual sentences. The sentential event recognizer extracts features from each sentence, and creates a feature vector for classification by a machine learning classifier. The classifier analyzes the features in the feature vector and uses these to assign an *event* or *nonevent* label to the corresponding sentence. Various types of features are useful to the classifier in making these decisions: lexical items such as *terrorist*, *assassination*, are good indicators of a terrorist event; similarly, semantic classes of words, such as diseases and symptoms, can be indicators of disease outbreak events; syntactic features and patterns from a parse tree can also be useful in recognizing event sentences. A more detailed description of the features appears in Section 3.3. Here, we focus on issues concerning the training of the sentence classifier.

One of the primary issues encountered in this component of the IE model is obtaining training data to train the classifier. The training data for this task consist of sentences with *event* and *nonevent* labels assigned to them. This then begs the question: *what constitutes an event sentence*? For many sentences there is a clear consensus among people that an event is being discussed. For example, most readers would agree that sentence (1) in Figure 3.3 is describing a terrorist event, while sentence (2) is not. However it is difficult to draw a clear line. Sentence (3), for example, describes an action taken in response to a terrorist event. Is this a terrorist event sentence? Precisely how to define an *event sentence* is not obvious.

Chapter 5 presents answers to these questions, through human annotation studies and data analysis. But, for the purposes of this discussion, let us take a peek into some of those findings. A clear consensus on event sentences is seen when a general time frame of the events is delineated. Our study finds that by breaking down the event description into smaller "incidents," and then deciding on the sentence labels based on their occurrence within the time frame achieves high agreement in humans. For example, the time frame for a disease outbreak event typically starts with the observation of specific symptoms in

(1) *Al Qaeda operatives launched an attack on the Madrid subway system.*
(2) *Madrid has a population of about 3.2 million people.*
(3) *City officials stepped up security in response to the attacks.*

Figure 3.3: Defining event sentences (example cases)

patients or victims, which is followed by a diagnosis, and in most cases ends with either a treatment, recovery or possible fatalities. Human readers tend to agree on the event sentence label for event descriptions in sentences that fall within this time frame.

Sometimes, however, the sentences refer to events in their entirety (e.g., *the bombing*, *last month's outbreak*, etc.), as opposed to subincidents within the events (e.g., *the car exploded*, *the windows blew out*, etc.). These mentions of events are usually accompanied by an event-specific detail (such as the location, victim, etc.) associated with the event. The human annotation study shows that there is again a strong agreement among human readers on the event sentence label for such sentences.

Our studies indicate that, to a large extent, human readers agree on what constitutes event and nonevent sentences. The human annotators achieve interannotator agreements of 0.77 and 0.72 Cohen κ for two types of events, respectively (see Chapter 5 for details of the study).

Human annotated sentences would be the ideal source of training data for the sentential event recognizer. The problem, however, is that human annotated data are expensive to create. Additionally, the need for such data restricts IE system portability by requiring new annotations from human experts for each new domain or task that the IE model is applied to. Consequently, this research explores alternate strategies for training the sentential event recognizer, without the need for human effort in sentence-level annotations.

3.2.1 Training with IE Answer Keys

Having humans annotate sentences with event and nonevent labels is an expensive proposition. One possibility for avoiding this annotation task is to approximate these using existing resources for IE. Rather than having human experts generate additional sentence-level annotations to create training data for the sentential event recognizer, leveraging existing resources to approximate these annotations allows for greater flexibility and portability of the model.

The supervised approach described here approximates event sentence annotations using standard IE data sets. Most current IE data sets typically consist of *answer keys* to enable the training and evaluation of IE systems. As described in Chapter 2, the data usually appear in the form of text documents with corresponding answer key templates. These answer key templates have been created by human experts, capturing the information pertaining to specific events of interest (e.g., terrorist events). An answer key template exists for each relevant event described in the document. For instance, in an IE task to extract event roles associated with terrorist event descriptions (*weapons*, *victims*, *perpetrators*, etc.)

the answer key template would contain the role fillers for the set of predefined event roles. The objective of IE systems trained for this data set is to identify this event role information in the given documents. The information in these answer key templates can be used to approximate the annotations of event sentences.

Given text documents and their corresponding IE answer key templates, the goal is to automatically assign *event* and *nonevent* labels to sentences in the documents using information from the templates. The answer key templates contain strings from the document that represent specific event roles within event descriptions. Mapping these event role strings to the document and identifying sentences that contain these strings can enable us to assign labels to these sentences. A sentence that contains a string from its corresponding answer key templates is considered an event sentence. All other sentences in the document, which do not contain a string from the templates, are labeled as nonevent sentences. The hypothesis underlying this approximate annotation scheme is that the event role strings are highly likely to be mentioned within event sentences as opposed to nonevent sentences.

Figure 3.4 illustrates this approximate annotation scheme using IE answer key templates.

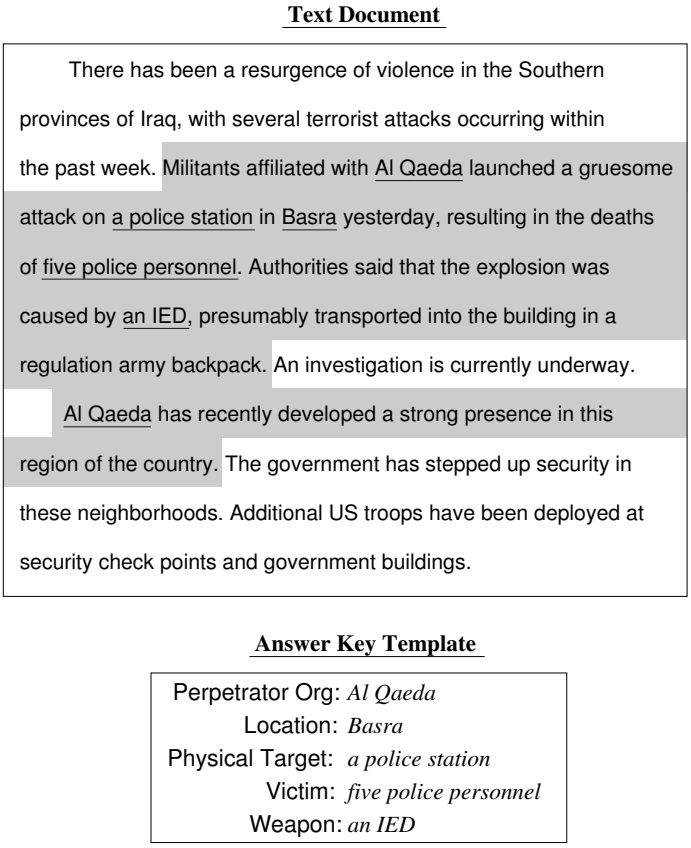


Figure 3.4: Approximate annotation with IE answer key templates

The text document in the figure describes a terrorist event, and the associated answer key template lists the event role fillers of that terrorist event. Any sentence in that document containing one of the listed role fillers is labeled as an event sentence. In this example, the sentences containing the strings *Al Qaeda*, *a police station*, *Basra*, *five police personnel* and *an IED* are considered event sentences (represented by the shaded sentences). The unshaded sentences do not contain any of the listed role fillers, and are considered nonevent sentences.

Observe, however, that these annotations can be noisy. In Figure 3.4, the second reference to "Al Qaeda" appears in a nonevent context: *Al Qaeda has recently developed a strong presence in this region of the country.* This sentence does not contain the description of a terrorist event. But because of the presence of the string "Al Qaeda," it gets incorrectly labeled as an event sentence. Similarly, sentences may incorrectly be labeled as nonevent sentences because they do not contain one of the listed role fillers. This can happen in cases where an event role is referenced as a pronoun, for example, and is not listed in the template. Thus, the noisy labels in the training data can be because of false positives as well as false negatives. In the grand scheme of things, however, these noisy labels do not occur too often. If the noise is acceptable for the classifier and answer keys are available, their use is preferred over the additional expense of sentence annotations by human experts.

Standard machine learning classifiers are trained on these data, and then used to identify event sentences in new unseen text. This research explores the use of discriminative Support Vector Machine (SVM) [119] classifiers, and generative Naïve Bayes classifiers. SVMs are discriminative classifiers that estimate a hyperplane separating the classes, based on given training data. Given a new test example, an SVM can then assign a class to this example by determining which side of the hyperplane it lies on. SVMs use kernel methods to estimate these hyperplanes, and have emerged as a popular machine learning approach for classification tasks. Naïve Bayes classifiers [76] employ a generative model for classification that estimates a probability of a class label for an example, based on the features associated with the example. The probability computation is simplified by making the assumption that all the features are independent of one another. Despite this simplistic assumption, Naïve Bayes classifiers have been shown to work reasonably well on real-world classification tasks. This research explores the used of these two classifiers for identifying event sentences.

A feature vector or training instance is created for each sentence in the training data. The features consist of lexical, syntactic and semantic properties of the input sentences. The list of features used by this model is described later in Section 3.3. The SVM and Naïve Bayes

classifiers are then trained on these training examples, to generate classification models for event sentences. Now given a new (unseen) sentence, these classification models can be used to determine if the sentence is an event or nonevent sentence. Just like the training example, a feature vector or test instance is created from the new (unseen) sentence, and the classification model is applied to this test instance to determine the assigned class.

3.2.2 Self Training with Document Labels

The previous section presented a simple supervised approach for training a sentential event recognizer using standard IE data sets. Even though the use of standard IE data sets eliminates the need for sentence-level human annotations, it still relies on answer key templates created by human experts. This section describes a training strategy that eliminates the need for answer key templates. The goal is to create a classifier that can identify event sentences in text, but does not depend on manually annotated sentence data or on IE answer keys to do so. The self-training procedure achieves this objective by requiring as input only a set of relevant and irrelevant documents for the domain, and a few seed patterns. Thus, rather than requiring sentence-level annotations, this approach uses document-level annotations, which are easier to obtain.

The document-level labels required in this training procedure indicate whether a given document contains the description of an event of interest. A document that has at least one event of interest is labeled as a *relevant* document. On the other hand, if a document contains no relevant events, it takes on the *irrelevant* label. Figure 3.5 shows an example of a relevant and an irrelevant document for the terrorism domain. Additionally, we want all of these documents from the same genre or "source," to prevent the classifier from picking up on stylistic or genre differences in making classification decisions. For example, if relevant documents about terrorist events were news articles chosen from American news sources, then the set of irrelevant documents should be other nonterrorism news articles from similar news sources. Ideally irrelevant documents that are semantically closest to the relevant documents are best suited for the irrelevant document set.

Observe that this results in an asymmetry in the training set. By definition, if a document is irrelevant to the IE task, then it cannot contain any relevant event information. Consequently, *all sentences in an irrelevant document must be nonevent sentences.* Thus, in this training procedure the irrelevant documents are used as a source of nonevent sentence training examples. In contrast, if a document is relevant to the IE task, then there must be at least one sentence that contains relevant event information. However, most relevant documents contain a mix of both event and nonevent sentences. These sentences are not

Relevant Document

There has been a resurgence of violence in the Southern provinces of Iraq, with several terrorist attacks occurring within the past week. Militants affiliated with Al Qaeda launched a gruesome attack on a police station in Basra yesterday, resulting in the deaths of five police personnel. Authorities said that the explosion was caused by an IED, presumably transported into the building in a regulation army backpack. An investigation is currently underway.

Al Qaeda has recently developed a strong presence in this region of the country. The government has stepped up security in these neighborhoods. Additional US troops have been deployed at security check points and government buildings.

Irrelevant Document

US Secretary of State Hillary Clinton says North Korea will face consequences for its "provocative and belligerent" actions towards its neighbors. As the UN Security Council discusses a response to North Koreas nuclear test, Mrs. Clinton reaffirmed that the US is committed to its allies Japan and South Korea.

Clinton spoke in Washington after North Korea's official news agency said Kim's government would no longer abide by the 1953 armistice tha ended the Korean War and may respond militarily to South Korea's participation in a US–led program to blockships suspected of carrying nuclear weapons or material for export. North Korea has raised tensions with its May 25th nuclear test.

Figure 3.5: Example of relevant and irrelevant documents

labeled, so the only information we have about the relevant documents is that one or more sentences within these documents are event sentences.

The seed patterns used in the training process are lexico-syntactic patterns like those found in existing pattern-based IE systems. The patterns define constraints on lexical, syntactic and semantic properties of segments of text. For the training procedure described here, we need the seed patterns to be such that they can reliably identify event sentences. In other words, if a segment of text meets the constraints specified by a seed pattern, we want to be able to reliably infer that the sentence (containing that text segment) is an event sentence. For instance, to build an IE system for terrorist events, seed patterns such as *"<subject> was kidnapped"* and *"assassination of <np>"* would be appropriate, since these phrases are strongly associated with terrorism and can reliably identify some terrorist event descriptions in text.

About 20 patterns manually created by a human expert can suffice for the training procedure. However, in this research even the seed patterns are semiautomatically generated from the relevant and irrelevant document sets. To obtain these seed patterns, an exhaustive set of extraction patterns is first generated from the relevant and irrelevant documents. In this work we use patterns based on the Sundance/AutoSlog [95] IE system.[3] So, we used the AutoSlog-TS system to generate the exhaustive set of patterns that literally extract every noun phrase in the document set. Of these, only patterns with a frequency greater than 50 were kept. These were then ranked by their probability of appearing in a relevant document, and the top 20 patterns were chosen as seeds. Because of ties, sometimes more than 20 seeds are obtained. In that case, patterns tied for last place are manually reviewed by a human expert to whittle the list down to 20.

Now that we have the seeds and the relevant and irrelevant document sets, we can proceed with the self-training procedure to train the sentential event recognizer. Figure 3.6 shows the self-training procedure, which begins with a handful of *seed patterns* to initiate the learning process. The patterns serve as a simple high-precision pattern-based classifier to automatically identify some event sentences. If a sentence contains one of the seed patterns, it is labeled as an event sentence, otherwise it is a nonevent sentence. In *iteration 0* of the self-training loop (shown as dotted arrows in the figure), the pattern-based classifier is applied to the unlabeled sentences to automatically label some of them as event sentences.

Next, an SVM [119] classifier is trained using these event sentences and an equal number of nonevent sentences randomly drawn from the irrelevant documents. The set of nonevent sentences is initially much larger than the set of event sentences, and this can cause the classifier to achieve a high accuracy by classifying most sentences as nonevent. Thus, the training data are artificially balanced to bias the classifier a little towards the event

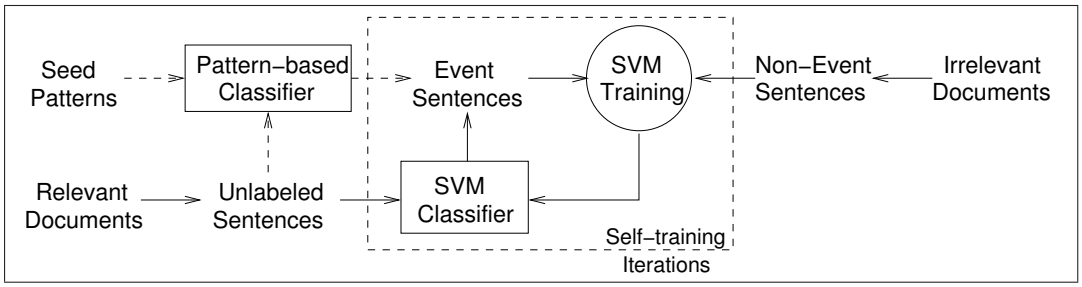


Figure 3.6: Self training for the sentential event recognizer

[3]The AutoSlog system defines about 21 types of patterns based on a shallow syntactic parse of the text. A description of these pattern types appears in Appendix C.

sentences, and to enable it to identify new event sentences. The feature set used by this classifier is the same as that used by the supervised classifier from the previous section. The features are based on the lexical, syntactic and semantic properties of each sentence, and will be described in greater detail in Section 3.3. The SVM is trained using a linear kernel with the default parameter settings. In a self-training loop, the classifier is applied to the unlabeled sentences in the relevant documents, and all sentences that it classifies as event sentences are added to the event sentences pool. The classifier is then retrained with all of the event sentences and an equal number of nonevent sentences to obtain a new classifier at the end of the self-training iteration.

The iterative self-training process is run for several iterations, and the SVM generated after each iteration is evaluated on held-out data. A more detailed evaluation of this iterative procedure appears in later chapters of this dissertation. The evaluation shows the performance of the classifier increasing for a few iterations, then leveling off (and then degrading). We measure SVM performance on held-out data to determine a stopping criteria for the self-training iterations. Based on this, we obtain our final classifier for use in the pipelined model.

This selected SVM classifier can be applied to the feature vectors generated from new (unseen) sentences to automatically assign event/nonevent labels to these sentences. We see that this classifier is more portable than the previous supervised classifier since it requires no sentence-level annotations. The document-level annotations and seed patterns used by this classifier are easier and cheaper to obtain, which enables us to quickly adapt this module to new domains and new IE tasks that do not have IE answer key templates available for training.

3.2.3 MIL Framework with Document Labels

The previous section describes a self-training procedure for training the sentential event recognizer using document-level annotations. This use of document-level annotations instead of sentence-level annotations closely matches a machine learning framework called *Multiple Instance Learning*. The Multiple Instance Learning framework has been studied by research groups as a way of reducing the amount of manual annotations required for training. This section explores the Multiple Instance Learning framework to eliminate the need for manual sentence-level annotations for training the sentential event recognizer.

Multiple Instance Learning [32] (MIL) is a learning framework developed specifically for cases where training examples are provided in sets or "bags" of examples. A bag has a positive label if any training instance contained within it has a positive label, and a bag

has a negative label if all training instances within it have a negative label. The classifier is trained using only the labels on the bags. These techniques were first developed in image analysis applications, where only a part of an image contained an entity of interest. Recently, these techniques have been applied to natural language applications, such as text categorization [3] and relation extraction [11].

Observe that the MIL paradigm fits our sentence classification task. We have annotations at the document-level for training a sentence-level classifier. This is analogous to the labels on the "bags" to classify individual instances. Each document is a "bag" of sentences (instances), where the document has a positive label (relevant) if any of its sentences has a positive label (event sentence). Similarly, a document has a negative label (irrelevant) if all of its sentences are irrelevant (nonevent).

A number of algorithms have been developed over the years for the MIL paradigm. Bunescu and Mooney [11] describe one such algorithm (sMIL) for training an SVM classifier. Their algorithm is specifically designed for the situation where the positive bags are sparsely populated with positive examples. A majority of the examples in these bags are negative examples. This is exactly the situation with our event/nonevent sentences, where the overwhelming majority of the sentences are nonevent sentences. Thus, this research investigates the use of this MIL algorithm for training the sentential event recognizer.

Bunescu and Mooney's sMIL algorithm for MIL builds on the Normalized Set Kernel (NSK) approach to MIL developed by Gartner et al. [40]. Gartner et al. develop an SVM kernel for MIL, which effectively combines the instances in the positive bags into a single training example, normalized by the L_1- or L_2-norm. An SVM is then trained on these combined examples. This approach for MIL implicitly assumes that a majority of the instances in the positive bags are positive, which is not true in many cases. Bunsecu and Mooney present the sMIL approach that modifies one of the constraints in the optimization expression of NSK, to favor smaller positive bags and sparse positive bags.

The freely available sMIL code by Bunescu and Mooney is used here to train an SVM classifier for sentence classification. The SVM is trained using document-level labels, and applied to feature vectors generated from new (unseen) sentences. We see that this classifier is portable just like the previous self-trained classifier, since it requires no sentence-level annotations. Additionally, this method uses no seed patterns, and the document-level annotations are easier and cheaper to obtain than sentence-level annotations. Again, like the self-trained classifier, the MIL framework enables us to quickly adapt the module to new domains and new IE tasks that do not have IE answer key templates available for training.

3.3 Sentential Features

The sentential event recognizers use trained classifiers to identify event sentences, based on features associated with each sentence. The set of features is automatically generated from the input text, and is divided into eight categories:

1. *Bag of words*: simple and popular type of feature used in many NLP applications. Each token in each sentence of the training data represents one such feature. These are all binary features, whose presence or absence in new test sentences is represented by boolean values. Despite their simplicity, such features work surprisingly well in text classification tasks.

2. *NP-head*: represents the lexical head of noun phrases. One such feature is generated for each noun phrase appearing in the training texts. These features are somewhat redundant with the bag of words, but are more informative because they are restricted to noun phrases.

3. *NP-sem*: represents the semantic class of the head of noun phrases. Certain semantic classes can be indicative of specific events, and as such could be useful for the classifier decisions. For example, weapon words can be indicative of terrorist events.

4. *Lexico-syntactic patterns*: represent the local context surrounding a noun phrase. These patterns are similar to extraction patterns used by existing IE systems, and they essentially capture the lexical and syntactic properties of the context around noun phrases.

5. *NP-feature*: represent specific features of a noun phrase that may indicate specific types of entities. Four types of characteristics of a noun phrase are used for this feature: (a) is the noun phrase a plural noun, (b) does the noun phrase contain a nationality (to identify phrases like *the German scientist*, *the Iraqi soldier*, etc.), (c) does the noun phrase contain a numerical modifier (for phrases like *ten people*, *134 civilians*, etc.), (d) is the noun phrase "extracted" by a communication verb pattern (e.g., *the newspaper* ***reported***). All of these characteristics of noun phrases can be suggestive of certain types of events.

6. *Named entities*: indicate names of people, locations or organizations mentioned in the sentences. These are useful for events where names of people or organizations are strong indicators of specific events.

7. *Sentence length*: two features are generated to flag sentences longer than 35 words or those shorter than five words. For many events, extremely long sentences or extremely short sentences can be informative for the classifier.

8. *Verb tense*: features representing the tenses of verbs used in each sentence, as determined by a part of speech tagger. These features are quite useful for events that are usually described in the past tense.

The feature set is automatically generated from the texts. Each feature is assigned a binary value for each instance, indicating either the presence or absence of the feature. The sentences in the training data are parsed by a shallow parser, and the syntactic and semantic features are generated by the Sundance/AutoSlog system [95]. We use the Sundance shallow parser to identify lexical heads and verb tenses, and we use its semantic dictionaries to assign semantic features to words. The Sundance system consists of semiautomatically created dictionaries of about 65,200 words, of which about 7,600 words appear in a domain-independent dictionary, 1,200 words appear in a terrorism-specific dictionary, and the remaining 56,400 words[4] appear in a biomedical dictionary. The nouns in this dictionary have semantic classes associated with them from a set of about 171 semantic classes. The verb tense features are also based on syntactic features assigned to words by the Sundance parser. Since this system is a shallow parser, it uses a relatively small tag set (compared to full parser such as the Collins parser [22]) for its part of speech tags and syntactic properties of words. We use its four tags — *PAST*, *PRESENT*, *FUTURE* and *PARTICIPLE* — as four binary verb tense features. The AutoSlog pattern generator [92] is used to create the lexico-syntactic pattern features that capture local context around each noun phrase. Named entities in the given sentences are identified by the Stanford NER Tagger [36]. We use the pretrained NER model that comes with the software to identify person, organization and location names. To make the feature set more manageable we apply a frequency cutoff of four to reduce the size of the feature set. Features appearing four times or less in the training documents are discarded. The resulting feature set is fed to the sentential event recognizers, to identify event sentences in text.

3.4 Localized Text Extraction

Having created a sentential event recognizer to identify event sentences in text, we now need a text extraction technique that can take advantage of event-labeled sentences.

[4] A large number of biomedical terms in this dictionary, like diseases and symptoms, have been obtained automatically from the UMLS [66] ontology.

The local context surrounding the candidate extractions can be more informative when considered in the perspective of the event sentences. For instance, if we are told that "Mr. Jackson died yesterday," the only information we glean from that phrase is that Mr. Jackson is dead. In other words, it informs us of the state of the entity (Mr. Jackson), but not how this state was attained. However, if we had additional information indicating that this phrase appeared in the context of a terrorist event description, we immediately make the inference that Mr. Jackson died as a result of the terrorist event. Thus, even though the phrase only provides us with state information about Mr. Jackson, combining that state information with the topic of discussion enables us to infer the role of Mr. Jackson in the event. In addition, since writers of books and articles (and speakers in conversations) rely on this skill of readers (and listeners) to make such inferences, direct indicators of role information are often omitted from text (and conversations). Thus, for an automated system to extract event role information accurately, it must be able to make such inferences. This component of the PIPER IE model aims to make such inferences about role information based on the event sentence information obtained from the sentential event recognizer, and on the state information about entities obtained from their local contexts.

Local contextual information has traditionally been used by IE systems in the form of extraction patterns. Extraction patterns learned and applied by existing IE systems essentially define constraints on the lexical, syntactic and semantic properties of the local context of words and phrases that are candidates for extraction. Following these approaches, in the PIPER IE model too we employ lexico-syntactic extraction patterns to encode the local context of plausible role fillers in the localized text extraction module. This model employs patterns based on the Sundance/AutoSlog [95] IE system, just like the seeds used by the self-trained sentential event recognizer in Section 3.2.2. A description of these patterns appears in Appendix C. Extraction patterns learned by the localized text extraction module are applied to the event sentences identified by the sentential event recognizer to extract event information.

The localized text extraction module now resembles a traditional pattern-based IE system, whose goal is to learn a set of extraction patterns for the given IE task. There are, however, several important differences between the patterns learned by traditional IE systems and those learned by the PIPER model. Patterns learned by traditional IE systems simultaneously decide whether a context is an event context and whether a word or phrase is a likely role filler. On the other hand, since the PIPER model already has a separate sentential event recognizer, the PIPER patterns are absolved of the responsibility to decide

whether a context is an event context. They only need to decide if a word or phrase is a desirable extraction within the event sentences. This completely changes the types of patterns that need to be learned by the PIPER model. Being liberated from the obligation of identifying event contexts, the PIPER patterns can be more relaxed. Patterns that would have been considered inconclusive or weak by a traditional IE system are now useful within the PIPER model. For example, a pattern like "<subject> *caught fire*" extracts entities that caught fire as the result of an accident or a deliberate act. However, within a terrorist event sentence, we could reasonably infer that the entity that caught fire was the target of the terrorist act. Thus, event sentence information enables the use of previously inconclusive local contexts for identifying event role fillers.

Now that it is clear that the localized text extraction module can employ weaker patterns for IE, we need to more precisely define the characteristics of the patterns we expect our model to learn. Once we know that we are in a domain-relevant area of text, patterns that simply identify words and phrases belonging to a relevant semantic class may be sufficient. In other words, a pattern that can reliably identify weapon words (like *bomb*, *explosive*, *rifle*, etc.) may be inconclusive by itself in determining the role of the weapons extracted, but when applied in a terrorist event sentence our model can infer that the weapons extracted are those used in the described event. So, the goal of the PIPER pattern learner is to learn patterns that extract *weapon-like* words or *victim-like* words or other plausible role fillers of specific event roles, based on the semantics of the extractions.

Two strategies for learning patterns for the localized text extraction module are described in the following sections — a pattern learner that uses IE answer key templates and a semantic affinity pattern learner that relies on a semantic dictionary. The sets of patterns learned by these approaches complete the PIPER model for IE. The patterns are applied by the localized text extraction module to event sentences identified by the sentential event recognizer to extract role fillers from these sentences.

3.4.1 Learning Patterns with IE Answer Keys

Given an extraction pattern, our goal is to determine if the pattern frequently extracts plausible role fillers for a specific event role. To do so we need to devise a metric that produces a numerical approximation of the extent to which a pattern exhibits this phenomenon. One simple approximation is to estimate the probability of a pattern extracting a plausible role filler for a specific event role. Such a probability could be computed by applying the pattern to a large corpus of text and getting frequency counts of extractions that are plausible role fillers and those that are not. The probabilities computed from these

counts can then be used to rank the patterns, and identify the best patterns for each event role.

One problem in the above strategy is that the notion of "plausible role filler" is somewhat subjective. A plausible role filler is a phrase or string that could potentially be a role filler for an event role. For example, *a toddler* could potentially be the *victim* of a terrorist event. However, it is unlikely to be the *perpetrator* of the event. Many plausible role fillers can be ruled out based on their semantics alone. For example, role fillers for the *weapon* used in a terrorist event are usually instruments or physical objects. Thus human words or phrases can usually be ruled out as plausible weapon role fillers.[5] Getting the desired frequency counts of such plausible role fillers is nontrivial. Having human experts annotate words or phrases with this information in the text corpus is a daunting task. We instead need to look to existing resources to approximate this annotation.

The pattern learner described here approximates these annotations using standard IE data sets, just like the supervised sentential event recognizer described in Section 3.2.1. As mentioned before, current IE data sets typically appear in the form of text documents with corresponding answer key templates. The answer key templates contain words or phrases filling specific event roles in the corresponding documents. Since the phrases in the answer key templates are role fillers, they provide implicit annotations for plausible role fillers in these documents. All occurrences of words and phrases in these answer key templates can be considered as plausible role fillers for the various event roles.

To rank patterns based on these answer key annotations, a ranking metric called *lexical affinity* is defined here. This metric measures the compatibility between a pattern and a specific event role based on a set of documents and their corresponding IE answer key templates. The lexical affinity between a pattern p and an event role r_k is determined by applying the p to the set of documents and computing the frequency $\mathrm{f}(p, r_k)$, which is the number of p's extractions that appear as event role r_k in one of the answer key templates. Then, the *lexical affinity* of pattern p with event role r_k is formally defined as:

$$\text{lex\_aff}(p, r_k) = \frac{\mathrm{f}(p, r_k)}{\sum_{i=1}^{|R|} \mathrm{f}(p, r_i)} \log_2 \mathrm{f}(p, r_k) \tag{3.1}$$

where R is the set of event roles $\{r_1, r_2, \ldots, r_{|R|}\}$. Lexical affinity is essentially the probability that a phrase extracted by pattern p is a plausible role filler for role r_k, weighted by

[5] There are exceptions, like "suicide bomber," who are human and could be considered a weapon.

the log of the frequency.[6] Note that it is possible for a pattern to have a nonzero lexical affinity for multiple event roles.

To generate extraction patterns for the localized text extraction module, the AutoSlog [95] extraction pattern generator is applied to the training corpus exhaustively, to literally generate a pattern for extracting every noun phrase in the corpus. The lexical affinity of each pattern is computed with respect to all possible event roles. This score is used to generate a ranked list of patterns for each event role, and the top-ranked patterns for each event role are selected for the localized text extraction module.

3.4.2 Learning Patterns with Extraction Semantics

The pattern learner described in the previous section relies on answer key templates to generate approximate annotations for plausible role fillers in text documents. This section, instead, describes a strategy that relies on semantic classes of words. The pattern learning strategy captures the relationship between extraction patterns and the semantic classes of their extractions in a metric called semantic affinity.

We developed a metric called *semantic affinity* [81] to automatically assign event roles to extraction patterns. Semantic affinity measures the tendency of a pattern to extract noun phrases that belong to a specific set of semantic categories. To calculate semantic affinity, we use a corpus of text documents and semantic dictionaries from the Sundance/AutoSlog [95] system. The semantic dictionaries are used to automatically assign semantic categories to words in a document. Alternatively, a resource such as WordNet [35] could also be used to obtain this information. Using these resources we compute the semantic affinity of each pattern to each event role, and use it as an approximation for the tendency of the pattern to extract plausible role fillers.

First, we define a mapping between the set of semantic categories in our dictionaries and the set of event roles defined by the IE task. For example, one role in the terrorism domain is *physical target*, which refers to physical objects that are the target of an attack. Most physical targets fall into one of two general semantic categories: BUILDING or VEHICLE. Consequently, we define the mapping "BUILDING, VEHICLE → Target." Similarly, we might define the mapping "HUMAN, ANIMAL, PLANT → Victim" to characterize possible victims of disease outbreaks. Each semantic category must be mapped to a single event role. This is a limitation of the approach for domains where multiple roles can be filled by the

[6]The formula used for lexical affinity is similar to other pattern ranking metrics used by previous IE systems [92, 124].

same class of fillers. However, sometimes a general semantic class can be partitioned into subclasses that are associated with different roles. For example, in the terrorism domain, both perpetrators and victims belong to the general semantic class HUMAN. But we used the subclasses TERRORIST-HUMAN, which represents likely perpetrator words (e.g., "terrorist," "guerrilla" and "gunman") and CIVILIAN-HUMAN, which represents ordinary people (e.g., "photographer," "rancher" and "tourist"), in order to generate different semantic affinity estimates for the perpetrator and victim roles. All semantic categories that cannot be mapped to a relevant event role are mapped to a special "Other" role. Appendix B presents the mappings that were used in this work.

To estimate the semantic affinity of a pattern p for an event role r_k, our model computes $\mathrm{g}(p, r_k)$, which is the number of pattern p's extractions that have a head noun belonging to a semantic category mapped to r_k. These frequency counts are obtained by applying each pattern to the training corpus and collecting its extractions. The *semantic affinity* of a pattern p with respect to an event role r_k is formally defined as:

$$\text{sem\_aff}(p, r_k) = \frac{\mathrm{g}(p, r_k)}{\sum_{i=1}^{|R|} \mathrm{g}(p, r_i)} \log_2 \mathrm{g}(p, r_k) \tag{3.2}$$

where R is the set of event roles $\{r_1, r_2, \ldots, r_{|R|}\}$. Semantic affinity is essentially the probability that a phrase extracted by pattern p will be a semantically appropriate filler for role r_k, weighted by the log of the frequency.[7] Note that it is possible for a pattern to have a nonzero semantic affinity for multiple event roles. For instance, a terrorism pattern like *"attack on <np>"* may have a semantic affinity for both physical targets, victims and locations.

To generate extraction patterns for an IE task, the AutoSlog [95] extraction pattern generator is applied to the training corpus exhaustively, so that it literally generates a pattern to extract every noun phrase in the corpus. Then for each event role, the patterns are ranked based on their semantic affinity for that role. The top-ranked patterns for each event role are used by the localized text extraction module to extract information from event sentences.

Table 3.1 shows the 10 patterns with the highest semantic affinity scores for four event roles in two domains — the terrorism domain and the disease outbreaks domain. In the terrorism domain, patterns are listed for the *weapons* and *perpetrator organizations* (PerpOrg) event roles. In the disease outbreaks domain, patterns are listed for the *diseases*

[7] The formula used for semantic affinity is similar to other pattern ranking metrics used by previous IE systems [92, 124].

Table 3.1: Top-ranked semantic affinity extraction patterns

Top Terrorism Patterns		**Top Disease Outbreak Patterns**	
Weapon	PerpOrg	Disease	Victim
<subject> exploded	<subject> claimed	cases of <np>	<# people>
planted <dobj>	panama from <np>	spread of <np>	<# cases>
fired <dobj>	<np> claimed responsibility	outbreak of <np>	<# birds>
<subject> was planted	command of <np>	<#th outbreak>	<# animals>
explosion of <np>	wing of <np>	<# outbreaks>	<subject> died
<subject> was detonated	kidnapped by <np>	case of <np>	<# crows>
<subject> was set off	guerillas of <np>	contracted <dobj>	<subject> know
set off <dobj>	<subject> operating	outbreaks of <np>	<# pigs>
hurled <dobj>	kingpins of <np>	<# viruses>	<# cattle>
<subject> was placed	attacks by <np>	spread of <np>	<# sheep>

and *victims* event roles. The patterns rely on shallow parsing, syntactic role assignment (e.g., subject (*subject*) and direct object (*dobj*) identification), and active/passive voice recognition, but they are shown here in a simplified form for readability. The portion in brackets (between < and >) is extracted, and the other words must match the surrounding context. In some cases, all of the matched words are extracted (e.g., "<# birds>"). Most of the highest-ranked victim patterns recognize noun phrases that refer to people or animals because they are common in the disease outbreak stories and these patterns do not extract information that is associated with any competing event roles.

3.5 Primary vs. Secondary Patterns

So far, our goal has been to find relevant areas of text, and then apply semantically appropriate patterns in those regions. Our expectation is that fairly general, semantically appropriate patterns can be effective if their range is restricted to event sentences. If our sentential event recognizer is perfect, then performing IE only on event sentences would be ideal. However, identifying event sentences is a difficult problem in its own right, and the sentential event recognizer is far from perfect. Consequently, one limitation of our proposed approach is that no extractions would be performed in sentences that are not deemed to be event sentences by the classifier, and this could negatively affect IE recall.

In addition, certain event-specific patterns can reliably identify event role fillers in text without the support of a sentential event recognizer. For example, the pattern "<subject> *was assassinated*" is a clear indicator of a terrorist event, and need not be restricted by the sentence classifier. If such a pattern matches a sentence that is classified as nonevent, then the classifier is probably incorrect. To take advantage of such patterns, our IE model was slightly modified to allow reliable patterns to be applied to all sentences in the text,

irrespective of the output of the sentential event recognizer. We refer to such reliable patterns as *Primary Patterns*. In contrast, patterns that are not necessarily reliable and need to be restricted to event sentences are called *Secondary Patterns*.

To automatically distinguish Primary Patterns from Secondary Patterns, we compute the probability of a pattern appearing in a relevant document, based on the relevant and irrelevant documents in our training set. We then define an upper conditional probability threshold θ_u to separate Primary patterns from Secondary Patterns. If a pattern has a high correlation with relevant documents, then our assumption is that it is generally a reliable pattern that is not likely to occur in irrelevant contexts.

On the flip side, we can also use this conditional probability to weed out patterns that rarely appear in relevant documents. Such patterns (e.g., "<subject> *held*," "<subject> *saw*," etc.) could potentially have a high semantic affinity for one of the semantic categories, but they are not likely to be useful if they mainly occur in irrelevant documents. As a result, we also define a lower conditional probability threshold θ_l that identifies irrelevant extraction patterns.

The two thresholds θ_u and θ_l are used with the two pattern learners to identify the most appropriate Primary and Secondary patterns for the task. This is done by first removing from our extraction pattern collection all patterns with probability less than θ_l. For each event role, we then sort the remaining patterns based on their lexical affinity score or their semantic affinity score for that role, and select the top N patterns. Next, we use the θ_u probability threshold to separate these N patterns into two subsets. Patterns with a probability above θ_u are considered to be Primary patterns for that role, and those below become the Secondary patterns.

3.6 Putting it Together

This chapter described a two-stage pipelined approach for IE that can make inferences by combining information from a sentential event recognizer and a pattern-based localized text extraction component. Several learning strategies with varying levels of supervision were described for each of the two components.

The first stage in the pipeline is the sentential event recognizer, whose task is to identify event sentences in text. This chapter presented three approaches for training this module with varying levels of supervision. The first approach (**Anskey**) requires maximum supervision in the form of IE answer key templates found in many standard IE data sets. The next two approaches use weaker supervision in the form of relevant and irrelevant documents. The self-training (**Self**) strategy uses a small set of seed extraction patterns

along with the document-level labels to train a sentence classifier. The multiple instance learning (**MIL**) approach views the documents with document-level labels as "bags" of instances and applies a standard MIL approach that was designed specifically for sparse positive "bags."

The second stage in the pipeline is the localized text extraction module, which uses a pattern-based approach to extract event role information from event sentences. This chapter presented two approaches with varying levels of supervision for learning patterns for this module. The first approach (**LexAff**) uses maximum supervision in the form of IE answer key templates to generate a pattern ranking, using a lexical affinity metric. The second approach (**SemAff**) uses weaker supervision from semantic dictionaries to generate a pattern ranking, using a semantic affinity metric.

The different variations of the two components of the PIPER model can be combined in several ways to form a single pipelined model. However, it makes better sense to combine components having equivalent levels of supervision to form the complete IE model. Based on this strategy for putting together these components, we have the following three configurations of the PIPER model for IE:

1. $\text{PIPER}_{\text{Anskey/LexAff}}$: This combination consists of the *Anskey* approach for the sentential event recognizer and the *LexAff* approach for the localized text extraction module. This combination in the pipeline requires the most supervision of the various configurations, since both components use IE answer key templates in their training strategies. As a result, the portability of this combination is limited to domains that have data sets consisting of IE answer key templates.

2. $\text{PIPER}_{\text{Self/SemAff}}$: This combination consists of the *Self* approach for the sentential event recognizer and the *SemAff* approach for the localized text extraction module. Since both components rely on weak supervision for training, the combined approach is weakly supervised, relying only on a set of relevant document, a set of irrelevant documents, a small set of seed patterns and semantic dictionaries. This further enables greater portability of this model to new domains and to new IE tasks.

3. $\text{PIPER}_{\text{MIL/SemAff}}$: This combination consists of the *MIL* approach for the sentential event recognizer and the *SemAff* approach for the localized text extraction module. Like the $\text{PIPER}_{\text{Self/SemAff}}$ model, this combination too relies on weak supervision for training, requiring only a set of relevant documents, a set of irrelevant documents and

semantic dictionaries. Therefore, this model is also easier to port to new domains and to new IE tasks.

To demonstrate the benefits of our pipelined approach, an extensive evaluation of these models is presented in Chapter 6. As we will see from the evaluation, the pipelined models perform well on two IE tasks. However, we also see that the sentential event recognizers employed in these models are not always perfect. The discrete decisions made in our pipelined model can prevent the localized text extraction component from overcoming these shortcomings of the sentential event recognizer, and can affect overall IE performance. The next chapter describes a probabilistic model to overcome the drawbacks of our pipelined approach.

CHAPTER 4

UNIFIED PROBABILISTIC MODEL

This chapter introduces a unified probabilistic model for IE, called GLACIER, that overcomes the drawbacks of the pipelined model for IE (PIPER) described in the previous chapter. GLACIER builds on the basic approach introduced in the pipelined approach. It combines a sentential event recognizer with a localized text extraction module to identify event role fillers in text. However, to balance the influence of the two components, GLACIER uses probabilistic models. Its decisions are based on a single joint probability that combines probabilistic estimates from the two components. This unified probabilistic model overcomes the drawbacks of the pipelined model, and results in an effective approach for IE. The details of this model are described in the following sections. Section 4.1 presents an overview of the unified approach, and discusses the characteristics expected of the components in this model. Section 4.2 covers the details of the sentential component, and discusses training strategies to generate probabilities of event sentences in text. Section 4.3 presents strategies for computing event role extraction probabilities, based on local contextual evidence. Section 4.4 then presents an overview of the contextual features used by the two components. Finally, Section 4.5 puts these components together into a single unified model, and discusses characteristics of the various configurations of this model.

4.1 Balancing the Influence of Components

The GLACIER model for IE presented here builds upon the PIPER pipelined approach described previously in Chapter 3. Like the pipelined approach, the GLACIER model also includes two primary components — a sentential event recognizer, and a component for identifying role fillers using local contextual clues. The GLACIER model shares PIPER's goal of using event recognition to better exploit local contextual clues. However, the GLACIER model has the additional goal of effectively balancing the influence of the two components, and using various types of contextual clues in this process.

The motivation behind this model is to allow the system to use reliable evidence from either of the two components while making extraction decisions. If the sentential event

recognizer is supremely confident about an event sentence, then much weaker local evidence can suffice in extracting event role fillers from the sentence. On the other hand, if the sentential event recognizer is not very confident about an event sentence, but there is strong local evidence for an event role filler, then in this case too the model should correctly extract the information with the given evidence. Likewise, if both components are partially confident about their information, the combination of the two may be strong enough to warrant an extraction. In this way, the GLACIER model aspires to more cleverly combine the pieces of information to make better extraction decisions.

A second motivation behind the GLACIER model is to combine information from several different clues in determining the roles played by phrases in a text document. For example, a contextual pattern such as "<subject> *died*" can be a small indication that the <subject> phrase could possibly be the victim of a disease outbreak. When added with a semantic clue, that the <subject> phrase is a human, can further strengthen the initial conjecture. Adding an additional lexical clue, that the head of the <subject> phrase is the lexical term "patient," can further increase the confidence of the system in its initial assessment. Finally, determining that this phrase appears within a disease outbreak event sentence can push its confidence beyond a threshold to enable the extraction of this phrase as the victim role filler of a disease outbreak. Thus, various clues can contribute to enable such an extraction. The GLACIER model achieves such a synergistic use of evidence, by generating probabilistic estimates based on the observed contextual clues.

The model uses the contextual clues within a sentence to generate a probabilistic estimate of the sentence being an event sentence. Similarly, it uses the local contextual clues around a phrase to generate a probabilistic estimate of the phrase being a role filler, *assuming that it appears in an event sentence*. The two probabilities are combined into a single joint probability, which is then used to identify role fillers in text. A more detailed description of this model appears later in this section.

Observe that this overall approach is fundamentally different from the PIPER model. In the PIPER model the two components are arranged in a pipeline, with the output of one fed as input to the other, and each component making discrete decisions independently of the other. The discrete decisions made by the two components lead to certain drawbacks that cause the model to overlook some event role extractions. The unified probabilistic approach of the GLACIER model can address these shortcomings and achieve better IE performance.

The first shortcoming of the PIPER model is that it makes discrete decisions about event sentences in the sentential event recognizer. This would not be as much of an issue if the

sentential event recognizer were flawless. However, since the classifier used in the sentential component can make mistakes, it can cause problems for the text extraction component. Once a sentence is incorrectly discarded as a nonevent sentence by the classifier, the model loses any chance of exploiting the weaker contextual clues for event role extraction in that sentence. Thus, we need a way to override the decision of the sentential event recognizer, if its confidence in the classification is low and there is strong evidence of an event role filler in the local context of a candidate phrase.

Another shortcoming of the PIPER model is that its localized text extraction component uses extraction patterns to represent the local context of candidate extractions. An examination of event descriptions in text documents indicates that many times other types of contextual clues can indicate event roles within event sentences. For example, the use of a weapon word in a terrorist event sentence can be strong evidence of it filling the weapon role in that event. Similarly, simply observing the word *assailant* in a terrorist event sentence can indicate the perpetrator of the event. Such additional lexical and semantic clues could enhance the role of lexico-syntactic extraction patterns in identifying event roles within the event sentences. Thus, we need our model to incorporate various type of clues into the model, and decide on event role fillers based on the aggregate contribution from all of these clues.

The GLACIER model presented here aims to overcome these shortcomings of the PIPER model by exploring a probabilistic framework for IE. The aim is to create a model that has the flexibility to make extraction decisions based upon strong evidence from the local context, or strong evidence from the wider context coupled with a more general local context. For example, some phrases explicitly refer to an event, so they almost certainly warrant extraction regardless of the wider context[1] (e.g., *terrorists launched an attack*). In contrast, some phrases are potentially relevant but too general to warrant extraction on their own (e.g., *people died* could be the result of different incident types). If we are confident that the sentence discusses an event of interest, however, then such phrases could be reliably extracted. The notion of "confidence" is captured by GLACIER in probability values. The use of these probability values and using them jointly in decision-making can overcome the drawbacks of the PIPER model resulting from its discrete nature.

The two types of contextual information are combined by the GLACIER model by incorporating them into a probabilistic framework. The model is designed for noun phrase

[1]There are always exceptions, such as hypothetical statements, but they are relatively uncommon.

extraction, and to determine whether a noun phrase instance NP_i should be extracted as a filler for an event role, GLACIER computes the joint probability that NP_i:

(1) appears in an event sentence, and

(2) is a legitimate filler for the event role.

Thus, mathematically, its decisions are based on the following joint probability:

$$P(EvSent(S_{NP_i}), PlausFillr(NP_i)) \tag{4.1}$$

where S_{NP_i} is the sentence containing noun phrase NP_i. The term $EvSent(S_{NP_i})$ indicates that sentence S_{NP_i} is an event sentence, and the term $PlausFillr(NP_i)$ indicates that the noun phrase NP_i in a plausible role filler for an event role. This joint probability estimate is based on the set of various contextual features F associated with S_{NP_i}. Including F in the joint probability, and applying the product rule, this joint probability can be split into two components:

$$\begin{aligned} & P(EvSent(S_{NP_i}), PlausFillr(NP_i)|F) \\ & = \quad P(EvSent(S_{NP_i})|F) * P(PlausFillr(NP_i)|EvSent(S_{NP_i}), F) \end{aligned} \tag{4.2}$$

These two probability components, in the expression above, form the basis of the two modules in the IE system — the *sentential event recognizer* and the *plausible role-filler recognizer*. In arriving at a decision to extract a noun phrase, the unified model for IE uses these modules to estimate the two probabilities based on the set of contextual features F, and makes its decisions based on input from both. Note that having these two probability components allows the system to gently balance the influence from the sentential and phrasal contexts, without having to make hard decisions about sentence relevance or phrases in isolation.

Figure 4.1 presents an overview of this model. Here, the sentential event recognizer is embodied in the probability component $P(EvSent(S_{NP_i})|F)$. This is essentially the probability of a sentence describing a relevant event. It operates on sentences obtained from a sentence boundary detector and assigns event sentence probabilities to them. Similarly, the plausible role-filler recognizer is embodied by the probability $P(PlausFillr(NP_i)|EvSent(S_{NP_i}), F)$. This component, therefore, estimates the probability that a noun phrase fills a specific event role, *assuming that the noun phrase occurs in an event sentence*. It operates on the noun phrases identified by a shallow parser, and assigns event role probabilities to each noun phrase. Finally, the probability estimates from the two components are combined into a single joint probability to make decisions about event role extractions.

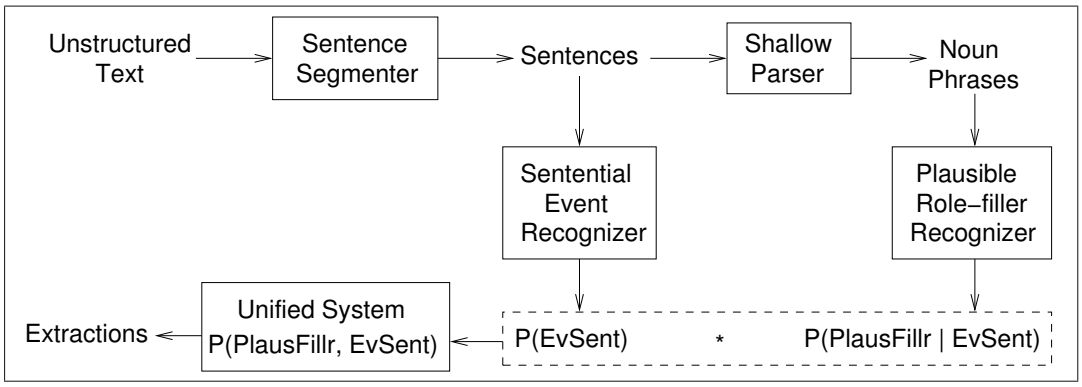


Figure 4.1: Overview of GLACIER — a unified probabilistic model for IE

Many different techniques could be used to produce these probability estimates. We need an approach that can analyze several contextual features and generate probability estimates for the two components based on these features. In the following sections, specific models for each of these components are discussed.

4.2 Obtaining Sentential Probabilities

As described in the previous section, the GLACIER model for IE contains a sentential event recognizer, which is similar to the sentential event recognizer employed in the pipelined model described in Chapter 3. The main difference here is that, instead of discrete event/nonevent classification, this component estimates the probability of each sentence being an event sentence. The task at hand for the sentential event recognizer is to estimate probabilities of sentences in a document discussing an event of interest. Given a new (unseen) text document, a sentence boundary detector segments the document into individual sentences. The sentential event recognizer analyzes various features associated with each sentence, and computes the probability of the sentence being an *event* or *nonevent* sentence. Similar to the features employed in the PIPER model, here too various lexical, syntactic and semantic features can be useful in generating the event sentence probability. A more detailed description of these features appears later in this section. Before that we focus on the strategies employed for computing these probabilities.

The task of estimating event sentence probabilities is similar to that performed by text classification systems. Text classification systems assign a class to each document based on various features associated with the document. Here we are dealing with the sentences, as opposed to entire documents. Also, instead of selecting a specific "class" for each sentence, we need a probability estimate of the "class," and not just a class label. Thus, we can explore text classification approaches for obtaining these probability estimates. Text

classifiers typically use supervised machine learning approaches to make decisions about text documents. Similarly, for the probability estimates too we explore the use of such machine learning classifiers.

In using a machine learning classifier for the sentential event recognizer, the primary issue we need to deal with is obtaining probability estimates from the classifier. Many classifiers make classification decisions for a data point (a sentence in our case) by estimating a score for each class assigned to the data point, and then selecting the class with the highest score. In probabilistic classifiers, such as the Naïve Bayes classifier, these scores are, in fact, the probabilities of the various classes. Thus, such classifiers could be used to directly estimate the probability of the event sentences in the given document. In other classifiers that generate a score for each class, the score could be normalized in a 0.0 to 1.0 range to get an approximate probability estimate. This research explores strategies to estimate sentential probabilities and discusses the associated issues.

Like the sentential event recognizer in the PIPER model (Section 3.2.1), here too we require training data to train the sentential classifiers. To avoid having human annotators create this training data by annotating sentences, we follow the procedure described in Section 3.2.1 to approximate this training data using existing IE resources. Supervised classifiers can be trained on such annotations to generate probability estimates for event sentences in text.

Using these annotations, the sentential event recognizer can now be trained to generate probability estimates for event sentences. This research explores the use of discriminative Support Vector Machine (SVM) [119] classifiers, and generative Naïve Bayes classifiers. A feature vector or training instance is created for each sentence in the training data. The features consist of lexical, syntactic and semantic properties of the input sentences. The classifiers are then trained on these training examples, to generate probability models for event sentences. Now given a new (unseen) sentence, a feature vector or test instance is created from the sentence, and the classifier is applied to this test instance to determine its probability of being an event sentence.

4.2.1 Naïve Bayes Event Recognizer

A Naïve Bayes classifier is a generative probabilistic model for classification that estimates the probability of each class based on the features of the instance. The class with the highest probability is assigned to the instance. In our case, each instance is a sentence S_{NP_i}, which can be assigned one of two classes — event sentence ($EvSent(S_{NP_i})$) or nonevent sentence ($\neg EvSent(S_{NP_i})$). Based on the set of features F associated with the sentence S_{NP_i}

the classifier computes the probabilities $P(EvSent(S_{NP_i})|F)$ and $P(\neg EvSent(S_{NP_i})|F)$. The class corresponding to the higher probability is assigned to the sentence.

Since Naïve Bayes classifiers estimate class probabilities as part of the classification decision, the sentential event recognizer can directly estimate $P(EvSent(S_{NP_i})|F)$ for the unified IE model. The probability is estimated by the classifier, using Bayes' rule:

$$P(EvSent(S_{NP_i})|F) = \frac{P(EvSent(S_{NP_i})) * P(F|EvSent(S_{NP_i}))}{P(F)} \tag{4.3}$$

In estimating $P(F|EvSent(S_{NP_i}))$, the classifier makes the simplifying assumption that the features in F are all independent of one another. With this assumption, the probability estimate becomes:

$$P(EvSent(S_{NP_i})|F) = \frac{1}{P(F)} P(EvSent(S_{NP_i})) * \prod_{f_i \in F} P(f_i|EvSent(S_{NP_i})) \tag{4.4}$$

where $P(F)$ is the normalizing constant, the product term in the equation is the likelihood, and $P(EvSent_{S(NP_i)})$ is the prior, which is obtained from the ratio of event and nonevent sentences in the training data. The features used by the model will be described in Section 4.4.

A known issue with Naïve Bayes classifiers is that, even though their classification accuracy is often quite reasonable, their probability estimates are often poor [33, 127, 68]. The problem is that these classifiers tend to overestimate the probability of the predicted class, resulting in a situation where most probability estimates from the classifier tend to be either extremely close to 0.0 or extremely close to 1.0. We observed this problem in our classifier too, so we decided to explore an additional model to estimate probabilities for the sentential event recognizer. This second model, based on SVMs, is described next.

4.2.2 SVM Event Recognizer

Given the all-or-nothing nature of the probability estimates that we observed from the Naïve Bayes model, we decided to try using a Support Vector Machine (SVM) [119, 56] classifier as an alternative to Naïve Bayes. One of the issues with doing this is that SVMs are not probabilistic classifiers. SVMs make classification decisions using on a *decision boundary* defined by *support vectors* identified during training. A decision function is applied to unseen test examples to determine which side of the decision boundary those examples lie. While the values obtained from the decision function only indicate class assignments for the examples, we used these values to produce confidence scores for our sentential event recognizer.

To produce a confidence score from the SVM classifier, we take the values generated by the decision function for each test instance and normalize them based on the minimum and maximum values produced across all of the test instances. This normalization process produces values between 0 and 1 that we use as a rough indicator of the confidence in the SVM's classification. We observed that we could effect a consistent recall/precision trade-off by using these values as thresholds for classification decisions, which suggests that this approach worked reasonably well for our task.

4.3 Estimating Role-Filler Probabilities

The plausible role-filler recognizer generates probability estimates for noun phrases being role fillers for specific event roles, *assuming they appear in an event sentence*. The plausible role-filler recognizer is similar to traditional IE systems, where the goal is to determine whether a noun phrase can be a legitimate filler for a specific type of event role based on its local context. Pattern-based approaches match the context surrounding a phrase using lexico-syntactic patterns or rules. However, most of these approaches do not produce probability estimates for the extractions. Classifier-based approaches use machine learning classifiers to make extraction decisions, based on features associated with the local context. Again, these classifiers are typically used to make classification decisions instead of generating probability estimates. Here we explore ways of generating probability estimates for event roles based on their local context.

Like the sentential event recognizer described in the previous section, here too we look into classifier-based methods to generate probabilities. Any classifier that can generate probability estimates, or similar confidence values, can be plugged into our model. We investigate probabilistic machine learning classifiers, such as the Naïve Bayes classifier, to estimate these probabilities in the phrasal role-filler recognizer.

In our work, we use a Naïve Bayes classifier as our plausible role-filler recognizer. The probabilities are computed using a generative Naïve Bayes framework, based on local contextual features surrounding a noun phrase. These clues include lexical matches, semantic features, and syntactic relations, and will be described in more detail in Section 4.4. The Naïve Bayes (NB) plausible role-filler recognizer is defined as follows:

$$\begin{aligned} & P(PlausFillr(NP_i)|EvSent(S_{NP_i}), F) \\ = \; & \frac{1}{P(F)} P(PlausFillr(NP_i)|EvSent(S_{NP_i})) \\ & * \prod_{f_i \in F} P(f_i|PlausFillr(NP_i), EvSent(S_{NP_i})) \end{aligned} \tag{4.5}$$

where F is the set of local contextual features and $P(F)$ is the normalizing constant. The prior $P(PlausFillr(NP_i)|EvSent(S_{NP_i}))$ is estimated from the fraction of role fillers in the training data. The product term in the equation is the likelihood, which makes the simplifying assumption that all of the features in F are independent of one another. It is important to note that these probabilities are conditioned on the noun phrase NP_i appearing in an event sentence.

Most IE systems need to extract several different types of role fillers for each event. For instance, to extract information about terrorist incidents a system may extract the names of perpetrators, victims, targets, and weapons. We create a separate IE model for each type of event role. To construct a unified IE model for an event role, we must specifically create a plausible role-filler recognizer for that event role, but we can use a single sentential event recognizer for all of the role filler types.

To train this classifier, we need phrase-level annotations with event role labels. Since our model operates on noun phrases and performs noun phrase extraction, the ideal training data for this component would be to have human experts annotate each noun phrase in the training documents with event role labels. Since we create a separate plausible role-filler recognizer for each event role, we also need separate sets of annotations for each event role. As a result of the multiple event roles in each event type, the manual annotations of noun phrases become even more complex and expensive than the sentence-level annotations. Thus, once again we turn to existing resources to approximate these noun phrase annotations.

The sentential components of both, PIPER and GLACIER, approximate the sentence-level annotations using standard IE data sets. The role fillers from the answer keys are mapped back to the corresponding documents, and any sentence containing one of the role filler strings is labeled as an event sentence. All other sentences are nonevent sentences. A similar strategy can potentially be applicable for annotating noun phrases as well. Identify all the noun phrases in the document that match a specific role filler (e.g., victim of a terrorist event) listed in the answer key template, and label these as "event role." All the remaining noun phrases are labeled as nonevent role. With this strategy we can obtain approximate event role annotations for noun phrases.

The annotations obtained from this process are "approximate" annotations, because not all of the role filler strings from the answer key templates mapped to phrases in the document appear in an event context. For example, the answer key template may contain the string "cows," a role filler for the victim of a disease outbreak. This string could appear

multiple times in a document, not all of which are in the context of the disease outbreak. But our approximate annotation scheme assumes that all mentions of "cows" in the document are role fillers for the victim event role. Thus, some of these get incorrectly annotated as positive examples.

Figure 4.2 illustrates this approximate annotation scheme using IE answer key templates for the "Perpetrator Org" event role. The text document in the figure describes a terrorist event, and the associated answer key template lists the role fillers of that terrorist event. Any noun phrase in that document matching the "Perpetrator Org" role filler string *Al Qaeda* is labeled as an event role noun phrase. In this example, there are two *Al Qaeda* noun phrases in the document, which get labeled as PerpOrg noun phrases. The remaining unshaded noun phrases are all labeled non-PerpOrg noun phrases.

This example also illustrates the "approximate" nature of these annotations. In Figure 4.2, only the first reference to "Al Qaeda" is a role filler for a terrorist event, since it specifically mentions the organization as being responsible for the attack. The second

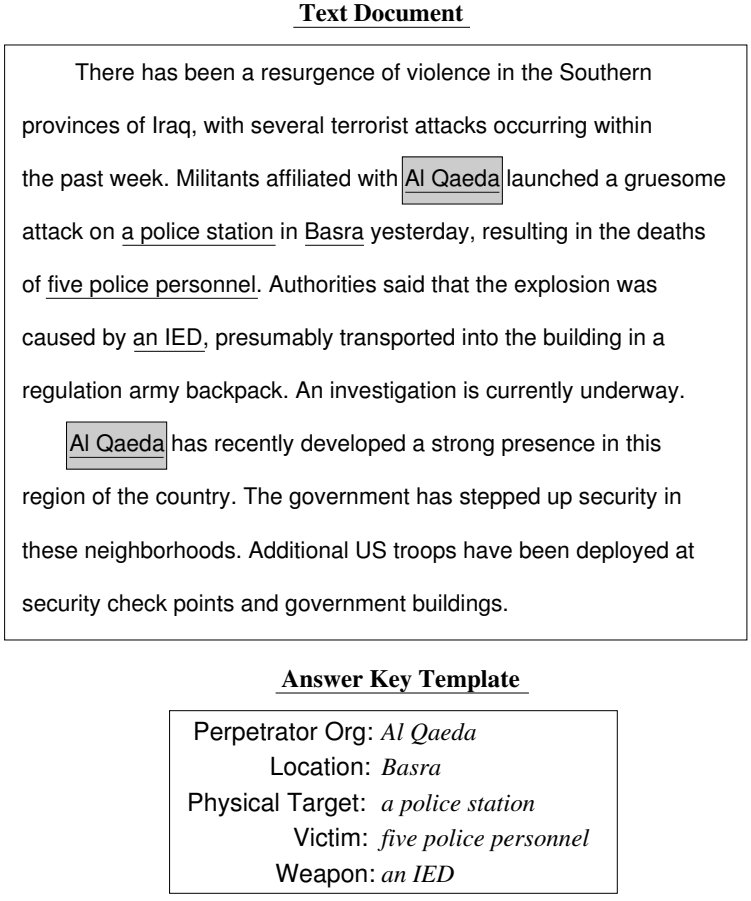


Figure 4.2: Approximate noun phrase annotation with IE answer key templates

reference to "Al Qaeda," however, is not a role filler for any event. It is only mentioned in a general fact about the organization. But our approximate annotation scheme assumes that both mentions of "Al Qaeda" in the document are role fillers in a terrorist event, and the second mention of "Al Qaeda" gets incorrectly labeled as an event role noun phrase. Thus, the noisy labels in the training data can be because of false positives such as these.

4.4 Contextual Features

We used a variety of contextual features in both components of our system, almost identical to those used in the PIPER model. The primary difference is that in the PIPER model, the features are used in the sentential component, while the localized extraction component relies on extraction patterns. Since we both the components of the GLACIER model rely on machine learning techniques, we have two sets of features generated for the GLACIER model. Most of the types of features generated for the plausible role-filler recognizer in this model, are reused in the sentential event recognizer, as will be described in this section.

Since many of the plausible role-filler recognizer features are also used later, these are described first. In this model, the plausible role-filler recognizer uses the following types of features for each candidate noun phrase NP_i: *lexical head* of NP_i, *semantic class* of NP_i's lexical head, *named entity tags* associated with NP_i and *lexico-syntactic patterns* that represent the local context surrounding NP_i. The feature set is automatically generated from the texts. Each feature is assigned a binary value for each instance, indicating either the presence or absence of the feature.

1. *Lexical head*: The lexical head of noun phrase NP_i is used as a feature in this model. This is based on our observation that certain lexical items (such as *terrorist*, *assailant*, *car-bomb*, etc.) can give away the event role (e.g., perpetrator or weapon of terrorist event) without the need for any other contextual information. One such feature is generated for each noun phrase appearing in the training texts.

2. *Semantic class*: The semantic classes of the head of each noun phrase NP_i is another useful feature. Many times the semantic class of a word (e.g., CRIMINAL) can be a strong indicator of an event role in an event context (e.g., perpetrator of a terrorist event). All the semantic classes of heads of noun phrases seen in the training texts are generated as features.

3. *Lexico-syntactic patterns*: To represent the local context around a noun phrase, lexico-syntactic patterns such as those used by IE systems are employed as features. The patterns capture the lexical properties and syntactic relations of the immediate local context around each noun phrase. A pattern generation component from an existing IE system is used generate an exhaustive set of lexico-syntactic patterns as features from the noun phrases in the training set.

4. *Phrase characteristics*: The PIPER model uses certain phrase characteristics as features for event recognition. Four types of characteristics of a noun phrase are used for this feature type: (a) a binary feature representing the plurality a noun phrase, (b) a feature for noun phrases containing nationalities as modifiers (to identify phrases like *the German scientist*, *the Iraqi soldier*, etc.), (c) a feature for noun phrases containing a numerical modifier (for phrases like *ten people*, *134 civilians*, etc.), (d) a feature representing noun phrases embedded in a communication verb pattern (e.g., *the newspaper* ***reported***). All of these characteristics of noun phrases can be suggestive of certain types of events, and result in four features added to the feature set.

5. *Named entities*: Many times names of entities can indicate role fillers (for example, the name of a city can indicate the location of an event). We use named entity features as indicators of such cases. Three types of named entities — person, organization and location names — are used as features to add three binary features to the feature set.

The *named-entity* features are generated by the freely available Stanford NER tagger [36]. We use the pre-trained NER model that comes with the software to identify person, organization and location names within each phrase. The syntactic and semantic features are generated by the Sundance/AutoSlog system [95]. We use the Sundance shallow parser to identify lexical heads, and use its semantic dictionaries to assign semantic features to words. The AutoSlog pattern generator [92] is used to create the *lexico-syntactic pattern* features that capture local context around each noun phrase. A more detailed description of these patterns can be found in Appendix C. Finally, a simple frequency-based feature selection method was used to reduce the size of the feature set by discarding all features that appeared four times or less in the training set.

In the sentential event recognizer we need features at the sentence-level, indicating the "eventness" of sentences. Since, noun phrases are subsets (or more precisely, substrings) of sentences, the features associated with noun phrases can also be used as features of sentences. Thus, our sentential event recognizer uses the all of the contextual features of

the plausible role-filler recognizer, except that features are generated for every NP in a given sentence. In addition to these phrase-based, sentence-level features are also used: *sentence length*, *bag of words*, and *verb tense*, all of which are also binary features.

1. *Sentence length*: Most commonly we find that extremely short sentences do not contain descriptions of events. Likewise, long sentences are more likely to contain an event description. Thus, one binary feature representing short sentences (shorter than five words), and one binary feature representing long sentence (longer and 35 words) are added to the feature set.

2. *Bag of words*: Since all of the words appearing in a document are usually used as features in text classification, here too the "bag-of-words" features are employed for identifying event sentences. One binary feature is generated for each word appearing in the training documents.

3. *Verb tense*: It is observed that events described in documents are frequently described in the past tense, since these description are of things that have already happened. Thus, the tense of the verbs used in a sentence could rule out certain sentences as event sentences. Thus, the tenses of verbs as determined by a part of speech tagger are added as features to the feature set.

The sentence length features and the bag of words features are based on the tokenization of sentences using the Sundance [95] NLP system. Similarly, the verb tense features are also based on syntactic features assigned to words by the Sundance NLP system. Since this system is a shallow parser, it uses a relatively small tag set (compared to full parser such as the Collins parser [22]) for its part of speech tags and syntactic properties of words. We use its four tags — *PAST*, *PRESENT*, *FUTURE* and *PARTICIPLE* — as four binary verb tense features. All of the different types of features obtained from the training data result in a large feature set for the sentential event recognizer. Here too, a frequency cutoff is applied to reduce the size of the feature set and make it more manageable. All features appearing four times or less in the training documents are discarded. With these features, the GLACIER model for IE can be trained and used for event-based IE.

4.5 Putting it Together

This chapter described a unified probabilistic approach for IE that can make inferences by combining information from a sentential event recognizer and a plausible role-filler recognizer component. This unified model overcomes the drawbacks of the two-stage

pipelined PIPER model by using probabilistic estimates instead of hard discrete decisions at each stage. This enables the model to gently balance the influence of the two components for IE. The different variations of the two components of GLACIER can be combined in several ways to form a single unified model. The components use machine learning approaches to generate probability estimates. Combining the variations of the two components, we have the following configurations of the GLACIER model:

1. $\text{GLACIER}_{\text{NB/NB}}$: This configuration consists of a Naïve Bayes sentential component, along with a Naïve Bayes phrasal component.

2. $\text{GLACIER}_{\text{SVM/NB}}$: This configuration consists of an SVM sentential component, used in combination with a Naïve Bayes phrasal component.

The components within this configuration are all trained with approximate annotations using standard IE data sets with answer key templates. We note that compared to some configurations of the PIPER model the GLACIER model does require a greater level of supervision in the form of IE answer key templates. However, an empirical evaluation in Chapter 6 will illustrate the benefits of joint decision-making in the the unified probabilistic approach presented here.

Although an extensive evaluation will be done later in this dissertation, to provide some insights into the GLACIER model, Figure 4.3 presents specific examples of extractions that are failed to be extracted by other localized models for IE, but are correctly identified by GLACIER. Observe that in each of these examples, GLACIER correctly extracts the underlined phrases, in spite of the inconclusive evidence in their local contexts. For example, in the last sentence in Figure 4.3, GLACIER correctly infers that the policemen in the bus are likely the victims of the terrorist event, without any direct evidence of this fact. This does not, however, provide any insights into why sentential information can be beneficial. The next chapter will present various analysis of event sentences to illustrate how sentential event recognition benefits IE.

THE MNR REPORTED ON 12 JANUARY THAT HEAVILY ARMED MEN IN CIVILIAN CLOTHES HAD INTERCEPTED A VEHICLE WITH OQUELI AND FLORES ENROUTE FOR LA AURORA AIRPORT AND THAT THE TWO POLITICAL LEADERS HAD BEEN KIDNAPPED AND WERE REPORTED MISSING.

PerpInd: HEAVILY ARMED MEN

THE SCANT POLICE INFORMATION SAID THAT THE DEVICES WERE APPARENTLY LEFT IN FRONT OF THE TWO BANK BRANCHES MINUTES BEFORE THE CURFEW BEGAN FOR THE 6TH CONSECUTIVE DAY — PRECISELY TO COUNTER THE WAVE OF TERRORISM CAUSED BY DRUG TRAFFICKERS.

Weapon: THE DEVICES

THOSE WOUNDED INCLUDE THREE EMPLOYEES OF THE GAS STATION WHERE THE CAR BOMB WENT OFF AND TWO PEOPLE WHO WERE WALKING BY THE GAS STATION AT THE MOMENT OF THE EXPLOSION.

Victim: THREE EMPLOYEES OF THE GAS STATION
Victim: TWO PEOPLE

MEMBERS OF THE BOMB SQUAD HAVE DEACTIVATED A POWERFUL BOMB PLANTED AT THE ANDRES AVELINO CACERES PARK, WHERE PRESIDENT ALAN GARCIA WAS DUE TO PARTICIPATE IN THE COMMEMORATION OF THE BATTLE OF TARAPACA.

Victim: PRESIDENT ALAN GARCIA

EPL [POPULAR LIBERATION ARMY] GUERRILLAS BLEW UP A BRIDGE AS A PUBLIC BUS, IN WHICH SEVERAL POLICEMEN WERE TRAVELING, WAS CROSSING IT.

Victim: SEVERAL POLICEMEN

Figure 4.3: Examples of GLACIER extractions

CHAPTER 5

EVENT SENTENCES AND EVENT ROLES

One of the key ideas presented in this research is the emphasis on recognizing relevant events in regions of text for improved extraction of event role fillers. The pipelined PIPER model for IE and the unified GLACIER model for IE both accomplish this with a sentential event recognizer for detecting event sentences in text. Automatically recognizing event sentences in text is nontrivial. Numerous questions arise in regards to this task: *What constitutes an event sentence? Can humans recognize event sentences effectively and consistently? How easy is this task for automated systems?* Answers to these questions bring out various issues that arise in recognizing event sentences, and also enable us to define the boundary between event and nonevent sentences. This chapter delves into the issues pertaining to event recognition, and starts (in Section 5.1) with an exploration of the role of event sentences within event descriptions. Section 5.2 then establishes a specific definition for event sentences, and investigates the agreement among human subjects on the task of recognizing such event sentences. In Section 5.3 we compare human annotations with approximate annotations from IE answer key templates. Finally, in Section 5.4, the relationship between event roles and event sentences is examined. This includes analyses of the data to quantify the potential benefits of sentential event recognition. The chapter concludes with a summary of these analyses in Section 5.5.

5.1 Event Descriptions in Text

To establish a clear definition for event sentences, we need to understand the types of roles they play within event descriptions. Each sentence written about a specific event typically serves a specific purpose in the event description. For example, some sentences describe the actions that took place at the scene of the event, while others describe properties or characteristics of entities at the scene. Knowing the purpose of a sentence within an event description can help more clearly separate event sentences from nonevent ones. We use such observations about event descriptions to establish a precise definition for event sentences. This section presents our observations regarding event descriptions that form the basis for

a human annotation study of event sentence recognition.

An analysis of documents containing event descriptions reveals several functional characteristics of the sentences in the given text. We find that some of these sentences are clearly distinguishable as event sentences. Typically, these are sentences that summarize the happenings in an event. We refer to these sentences as *event incident* sentences. Such event incident sentences attempt to paint a picture of the event by describing actions and characteristics of entities as observed during the occurrence of the event. Likewise, sentences that are completely unrelated to the event are easily identified as nonevent sentences. We refer to these (unsurprisingly) as *not relevant* sentences. Apart from these two types of sentences, we observe a number of sentences that lie on the boundary between event and nonevent sentences. These sentences contain information that is somewhat peripheral to the event description. We further categorize these border cases into three functional categories — *event consequence*, *event precursor* and *incidental information*. These functional categories are further discussed in the following paragraphs.

Event incident sentences assume that an event comprises of multiple incidents that take place chronologically in the real world. Each of these incidents contributes to the event as a whole. For example, a disease outbreak event starts with the observation of specific symptoms in patients or victims, which is followed by a diagnosis, and in most cases ends with either a treatment, recovery or possible fatalities. Event incident sentences describe the various incidents that comprise the event of interest. Note that even though the incidents occur chronologically in the real world, they need not be described in the same order in the event description. These sentences also vary in their level of detail, depending on the intended audience and the nature of the event description. For example, when an event is described as a fleeting reference in a single sentence, usually very little detail of that event is provided. On the other hand, complete stories exclusively about an event go into a lot more detail about the event. In addition to these incident descriptions, sometimes sentences refer to the event as a whole (e.g., *the outbreak*, *the attack*, etc.) with some specific detail about the event (such as the location, the time it took place, a victim of the event, etc.). These are also considered as event incident sentences.

The *event precursor* and *event consequence* sentences are related but are somewhat peripheral to the actual event description itself. These sentences essentially describe incidents that take place before or after an event, and are causally related to the event. Event precursors are incidents that are thought to be the cause of the described event. Similarly, event consequences are incidents that take place after an event and are considered to be

caused by the event. For instance, a government organization may step up security in a region as a result of a terrorist attack. This government action is a consequence of the attack. Similarly, if the terrorist attack is in retaliation to an earlier incident, then that earlier incident is an event precursor of the terrorist attack. The temporal or spatial nearness of precursors and consequences to the event incidents can make it difficult for humans to agree on event/nonevent labels for such sentences, and thus are considered to be boundary cases between event and nonevent sentences.

The third type of sentences that lie on the periphery of event descriptions are the *incidental information* sentences. These sentences do not pertain directly to the event description, but usually contain additional information about the entities mentioned in the event description. Such sentences occur in news stories because, frequently to capture reader interest, writers include intriguing facts or properties of the location, persons or objects involved in the event. Sometimes, interesting facts pertaining to the date or time at which the event took place are described in the document. Usually, these sentences do not relate specifically to the event or to the consequences of the event, but only are details of the persons, objects, dates or locations playing a part in the event. For instance, consider the following sentences:

> *The explosion blew out all the windows of an empty building. The building was constructed in 1901, and had been scheduled for demolition next month.*

The second sentence presents some information (potentially interesting to the reader) pertaining to the building mentioned in the first sentence. The second sentence is a factual statement and does not specifically relate to the explosion incident mentioned in the first sentence. We consider such sentences to be *incidental information* sentences. Additionally, sentences that refer to the event as a whole, but provide only "meta-information" about the event (i.e., no event-specific detail) are also considered incidental information sentences. For example, a sentence such as:

> *This is the third such bombing in Bogota in the past three months.*

does not provide a specific detail about the bombing itself, but puts it into perspective with other bombing events. As such, this sentence is also considered *incidental information*.

Finally, sentences that do not pertain to an event of interest at all are considered *not relevant* sentences. Since these sentences are unrelated to the event of interest, these present little ambiguity by way of event/nonevent sentence labels, and are usually easy to recognize as nonevent sentences by human readers.

Figures 5.1 and 5.2 illustrate the five types of functional sentence categories in documents describing two events. Figure 5.1 shows the sentence types in a story about a terrorist event. There are three main regions (groups of sentences) of text that describe core incidents of the terrorist event (event incidents). Another two regions of text then provide some additional information (incidental information) about entities or objects mentioned in the event incident sentences — *the two US missionaries*, *the FAL terrorist organization* and the *message* left by the terrorists. Towards the end of the document we come across two regions providing incidental information about the event itself, and one region describing the consequences of the event. A sentence that is not relevant to the event description is seen at the end of the document. A similar analysis of a disease outbreak event is shown in Figure 5.2. The first region in this document lists the event incident sentences describing the core happenings of the event itself. In the remainder of the document, two regions then contain further information (incidental information) about the *Tularemia* disease, and one region of text describes the consequences of the disease outbreak event. At the bottom of the document, we see a sentence that is not relevant to the disease outbreak event.

Such analysis of documents can enable us to take the first steps towards a clear definition of event sentences. In addition, they can also give us some insights into the nature of the event description. For example, if a document contains only one or two event incident sentences surrounded only by sentences that are not relevant, then the description is most likely a fleeting reference to an event. Similarly, if the document is densely populated by event incident sentences, it most likely contains a blow-by-blow account of the event. Most documents, however, lie somewhere in between these two extreme cases. Such analysis enables a better understanding of the functional roles of sentences, which is used in setting up a human annotation study, described next.

5.2 Human Annotation Study

The Piper and Glacier models for IE described previously in this dissertation both contain components designed to identify event sentences in text, and aim to improve IE performance using this information. The knowledge of event sentences enables the system to make inferences about event roles, even with weak local evidence. Before evaluating the two IE models in their entirety, it would be interesting to evaluate the performance of the componentsin isolation, and then evaluate their impact on the aggregagte IE models. To measure their individual performance, we need to determine if the event sentences identified by these components match a human expert's notion of event sentences.

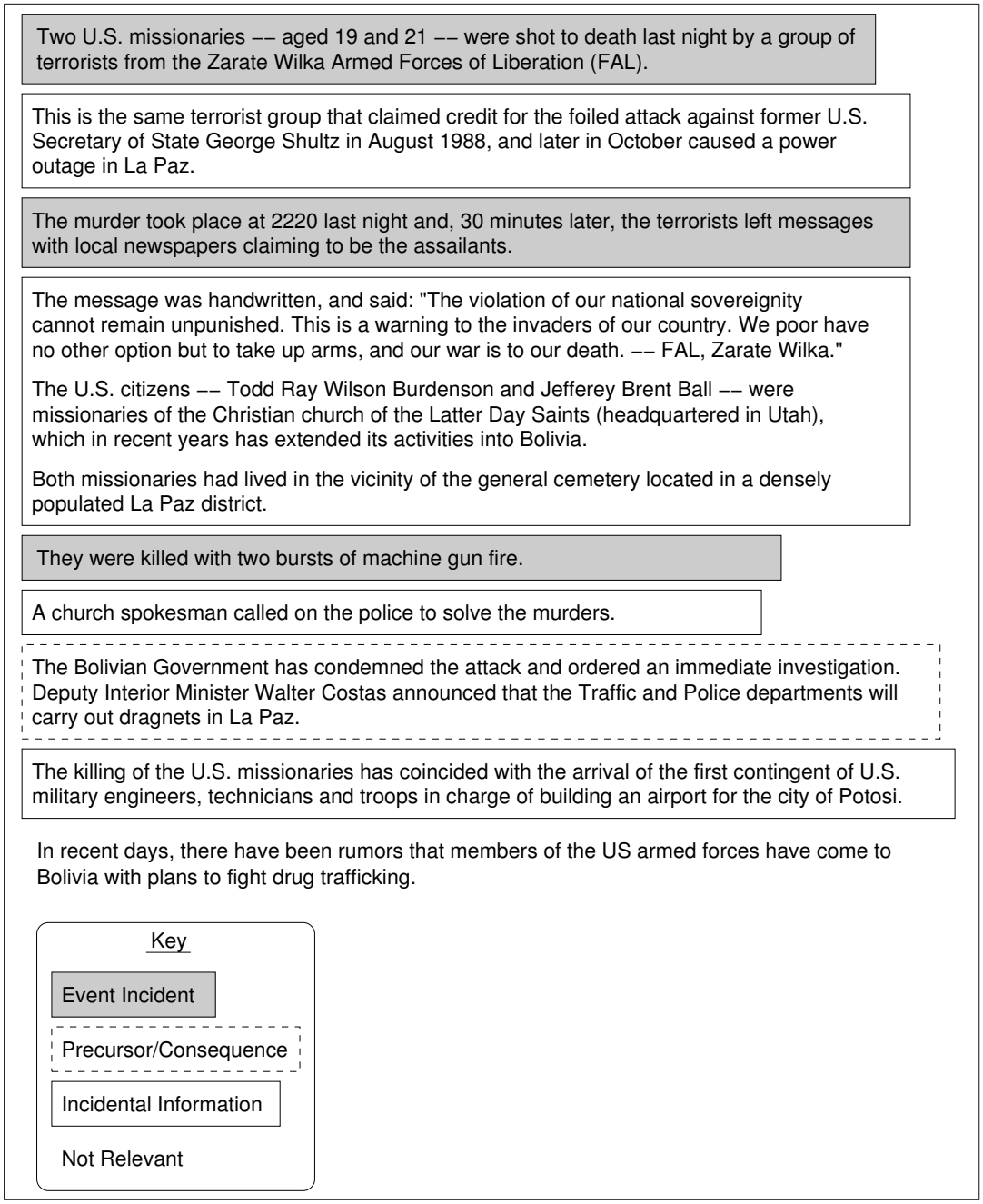


Figure 5.1: Sentence types in a terrorist event description

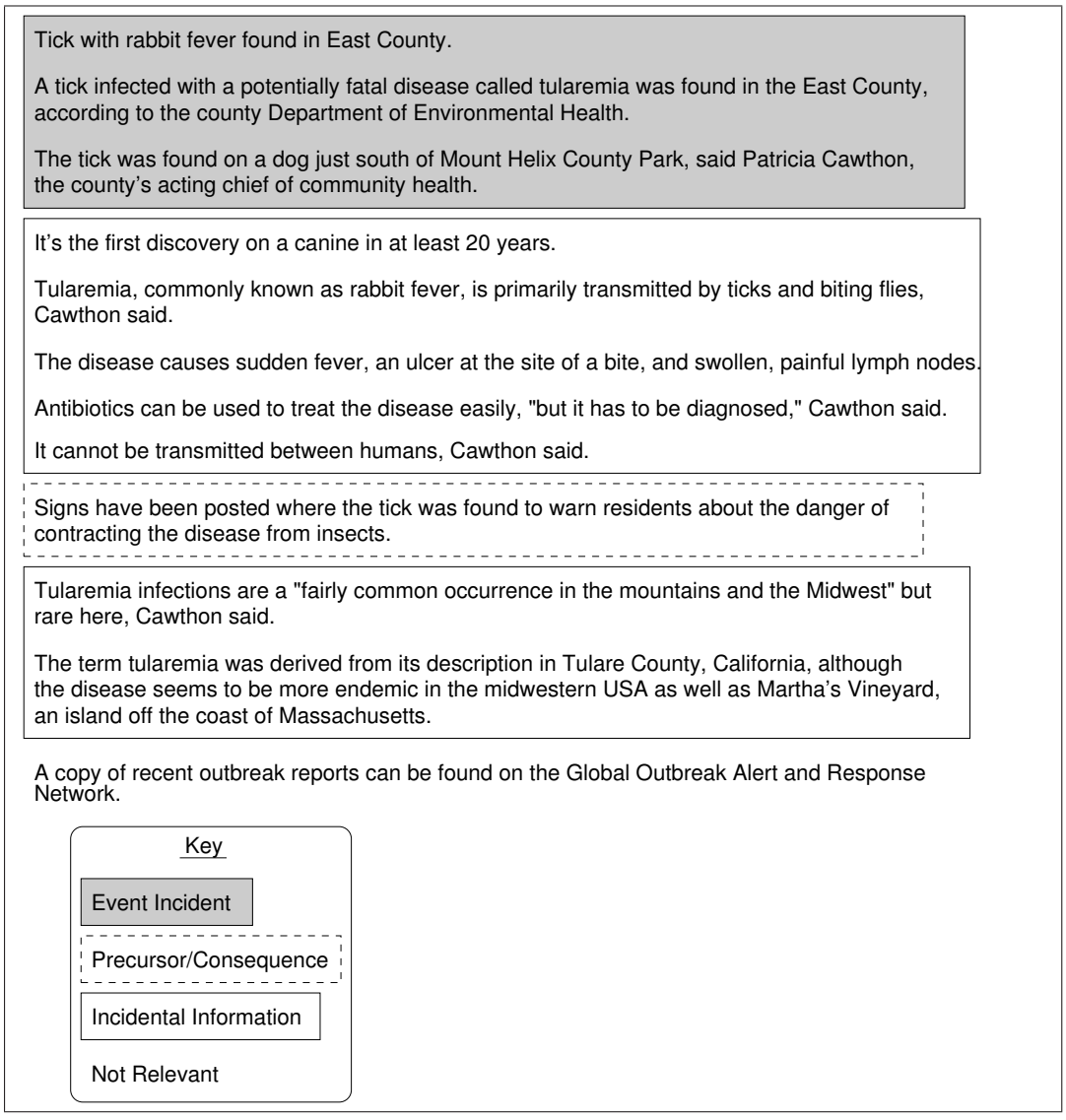


Figure 5.2: Sentence types in a disease outbreak description

Thus, we developed a gold standard data set through a human annotation study. The annotation study also provides us with insights into the feasibility of this task. This section describes the human annotation study, including the guidelines that were provided to the human annotators, the details of the annotation software used, and the agreements that were obtained on this task.

5.2.1 Annotation Guidelines

An event is broadly defined as an occurrence, incident or happening at a given place and time. With this broad definition of events, a single text document can simultaneously contain descriptions of many different events. Almost every action or incident described in the document could be considered as a separate event. However, for the purpose of this research our interest specifically is in events pertaining to the given IE task. The objective of our IE models presented in this research is to recognize descriptions of those specific events whose role fillers we need to extract from the text documents. Thus, our definition for the events of interest is obtained from the instructions associated with the given IE task. For example, the MUC-4 task for extracting role filler for terrorist events, defines terrorist events as "*violent acts perpetrated with political aims and a motive of intimidation.*" Our sentential event recognizers must then identify event sentences using this definition of an "event" for this given IE task. The goal of this annotation study is to determine if humans agree on event sentences based on such a definition of an event.

Simply defining an event sentence as one that contains the description of a relevant event is rather vague. Given this simple definition of event sentences, we could imagine a wide variation in the way people assign labels to sentences. One person might decide to include consequences of events in an event description, while another person might choose to leave those out. Similarly, one person might consider incidental information as part of an event description, while others might discard such information as being not event-related. Wide variations in human judgements on a task make it impossible to evaluate an automated system, since there is no established ground truth.

The problem is that a simple definition of event sentences leaves a lot of decisions for the human annotators, which can be influenced by their individual biases, preferences and contexts. What we need is a more specific definition for event sentences that matches the requirement of our IE task. Accordingly, we describe here the guidelines for labeling event and nonevent sentences in text, to develop a gold standard data set. The goal is to have guidelines that (a) are specific enough that internal biases of the human experts do not affect their annotations very much, (b) are general enough to be adaptable to different

event types, and (c) capture the essence of "event sentences" as required for the IE task.

To incorporate these requirements into the annotation guidelines, we use the sentence types that were observed in Section 5.1. The sentence types encompass five different types of information — *event incidents*, *event precursors*, *event consequences*, *incidental information* and *not relevant*. By definition, the event incident sentences describe the incidents comprising the event of interest, and thus are likely to be event sentences. Similarly, the not relevant sentences, by definition, are sentences that do not pertain to an event of interest. Therefore, these sentences are likely to be nonevent sentences. The remaining three types — event consequences, event precursors and incidental information — contain information that is somewhat peripheral to the event descriptions and, therefore, can be considered the borderline cases.

To establish the formal annotation guidelines based on the above five sentence types, we relied on the expertise of human annotators. The annotation study (described later in Section 5.2.3) consisted of a "training phase," during which the annotators learned the annotation scheme and got familiar with the annotation interface. Part of this training phase was also used to obtain feedback from the annotators to refine the annotation guidelines. The annotators were initially asked to annotate sentences with the five sentence type labels mentioned above. From these initial annotations we found that, while the agreement of the annotators on event incident and not relevant sentences was quite high, their agreement on the remaining three sentence types was quite poor. Additionally, we found that the event precursor, event consequence and incidental information sentences were often confused with the not relevant sentence type. Thus, our final annotation guidelines for event sentence labels were based on the event incident sentence type described earlier, and the nonevent sentence labels were based on the remaining four sentence types. The exact annotation guidelines that were developed and provided to the annotators are presented verbatim in Appendix A and are briefly summarized in the following paragraphs.

Based on these established annotation guidelines, event sentences now follow the definition of the event incident sentences and assume that an event is decomposed into a sequence of smaller incidents, which lie on a chronological time frame. The event is then described by recounting the noteworthy incidents that occurred during this event time frame. Any incident descriptions from this time frame (and pertaining to the event) are labeled as event sentences. Additionally, sentences that refer to the event as a whole and contain event-specific details are also labeled as event sentences.

Similarly, based on the final annotation guidelines, the remaining four sentence types

comprise the nonevent sentences. Reiterating the properties of these, the event precursor and event consequence sentences describe incidents that are typically outside the primary event time frame (but causally related to the event). Similarly, the incidental information sentences describe facts or characteristics of specific entities mentioned in the event or provide additional context for the event itself. These sentences may contain references to event roles, but not in the context of the event. Finally, the not relevant sentences do not contain any information directly pertaining to the event, and are also labeled as nonevent sentences.

All of the guidelines discussed so far assume a single event description per document. However, in practice, a single document may contain descriptions of multiple events. In other words, it is quite common to see documents containing descriptions of two or three separate terrorist events. In a full-fledged IE task, it is important to separate the event roles of the various events described. While the goal of this research is not to separate the event roles extracted, we do hope that the annotations created here can be useful in the future for the full-fledged task. Therefore, in addition to the event/nonevent annotations assigned to sentences, the annotators are asked to attach one or more event identifiers to each event sentence annotated by them. An event identifier indicates a specific event description within a document. In documents containing more than one event description, the event sentences are mapped to the corresponding event descriptions through the event identifiers. So, it is the annotators' task to identify event sentences belonging to the description of the same event (represented by an identifier) and those belonging to separate events. The annotation guidelines provide instructions about assigning event identifiers to event sentences.[1]

In assigning event identifiers to event sentences, the annotator first needs to identify the number of different event descriptions in any given document. This is a nontrivial task, since the annotators now need to also interpret the event definition provided for the IE task in addition to the event sentence guidelines provided here. To simplify the task for the annotators, we use the IE answer key templates associated with a document to automatically determine the number of event descriptions in a document. An IE answer key template can be considered to be an "event summary" of an event description in the document. Thus, the number of answer key templates associated with a document is an indicator of the number of event descriptions in that document. We automatically generate

---

[1]Nonevent sentences, by definition, cannot be part of an event. Therefore, they are not assigned event identifiers.

event identifiers and brief event summaries using these answer key templates, and require the annotators to select only one or more of these identifiers for each event sentence labeled by them. Section 5.2.2 shows how this is done within the annotation interface provided to the annotators.

5.2.2 Annotation Interface

In addition to the guidelines defining event and nonevent sentences, human annotators are provided with an annotation interface to ease the annotation task for them. The purpose of the annotation interface is threefold. Firstly, by having a point-and-click interface the annotations can be done much faster. Secondly, it provides a single standard input/output format for storing and accessing the annotations, and provides a single standard method for segmenting the given documents into sentences for annotation. Finally, it enables the annotators to save partially annotated data and return to the annotation task as their schedule permits. In addition, the annotation software also contains an agreement scorer that reads in multiple annotations of a single data set and generates a summary of the annotator agreements.

The annotation interface created for this task is a Graphical User Interface (GUI) that can load a data set consisting of text documents and enable the annotation of sentences in these documents by simple point-and-click interactions with this interface. Figure 5.3 shows a picture of this interface in action. The interface consists of three main parts. On the extreme left of the GUI is the list of text files in the loaded data set. The middle of the GUI contains sentences of the current document being annotated. The extreme right contains the status and annotations of the current sentence under consideration. The figure shows a data set of disease outbreak events loaded in the GUI, and shows one partially annotated document in this data set. The application also contains progress bars to track the annotation progress on the current file and on the entire data set as a whole.

To start off the annotation process the annotator selects *Setup Annotations* in the *Annotation* menu, a drop-down menu at the top of the interface. This brings up a dialog that requests four pieces of information:

1. *Domain*: a Sundance domain representing the type of event. The interface relies heavily on the Sundance/AutoSlog system [95], and uses its event domains for the event type to be annotated. For example, the *bio* domain in Sundance represents disease outbreak events, and the *terror* domain represents terrorist events. A "domain" in Sundance essentially loads domain-specific dictionaries and other document format

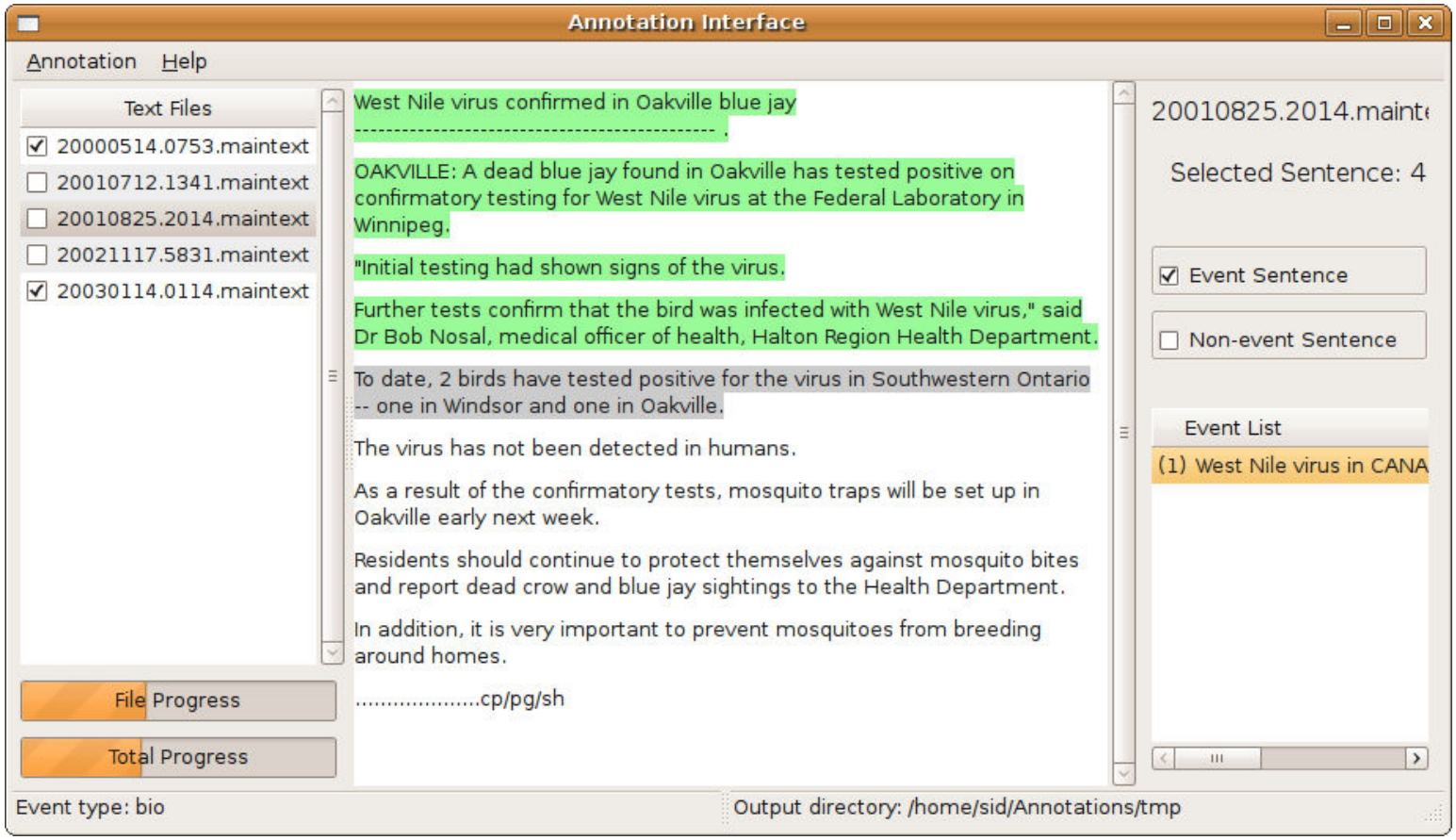


Figure 5.3: Graphical user interface for the annotation task

rules (e.g., skipping appropriate document headers) for the given event type. Note, however, that in this interface Sundance is used only for sentence segmentation, and therefore does not use the domain-specific dictionaries.

2. *Output Directory*: the directory where the annotations will be written. One annotation file is created in this directory for each input document in the data set. This directory can be empty initially. If the directory already contains annotations for the data set, those are loaded into the interface. Any other files appearing in the directory are ignored.

3. *SList of Text Files*: the data set to be annotated is specified in an "SList" file. SList files are used by the Sundance system to represent a set of files or documents. An SList file contains a path to the directory containing the files, followed by a list of file names. Here, the SList file points to a set of text documents to be annotated by the annotators.

4. *Answer Key*: the IE answer key templates corresponding to the text files. This is an optional field also specified as an SList file representing a set of IE answer key template files. The purpose of these IE answer keys in the annotation process will become clear later in this section.

With this information, the list of documents in the data set is loaded into the list on the left side of the GUI. The text area in the middle, and the annotation information to the right are initially empty or disabled. The status bar at the bottom initially lists the event type (from the domain field) and the output directory that the annotations will be written to.

As indicated in the previous paragraph, if the output directory already contains annotations for the loaded data set, then these annotations are loaded into the interface. This feature is meant to enable the annotators to save partially annotated data and come back to it later as time permits (without having to leave the GUI or their computer running). For each document in the data set, the corresponding annotations in the output directory are stored in a file with the name of the document with a ".ann" appended to it. For instance, the annotations corresponding to a document in a file named "MUC4-TST4-0039" will appear in a file named "MUC4-TST4-0039.ann" in the output directory. Thus, when the data set is loaded, the interface looks for the corresponding annotation files in the output directory. If the annotation files are found there, then annotations for those documents are

loaded into the interface. If no corresponding annotation files are found, then the documents are assumed to be unannotated.

The progress indicators below the document list enable the annotators to keep track of their work, and get an estimate of how much remains. Thus, when the documents are initially loaded, if no annotations already exist in the output directory, these progress bars are initialized to zero. On the other hand, if partial or complete annotations for the data are loaded from the output directory, then the progress bars reflect the extent of the work already completed. The progress on a data set that is loaded for annotation is, therefore, immediately visible in the "Total Progress" progress bar (the second progress bar) below the document list.

Having loaded the data set into the annotation interface, the annotator can now begin the annotation process. To start, the annotator selects from the list of documents, a document to be annotated. Double-clicking on the document name loads that document into the text area in the middle of the GUI. The document is sentence-segmented using the Sundance system, and the sentences are displayed in the text area. Having all the sentences of the document be visible together in the text area, enables the annotation of these sentences in the context of the entire document, rather than in isolation. The goal of the annotator now is to assign labels to each of these sentences.

To assign a label to a sentence, it first needs to be selected from the list of sentences. Clicking on a sentence selects it for annotation. A selected sentence appears with a gray background in the GUI. As soon as a sentence is selected, the state of the annotation status and controls on the right side of the GUI are enabled and updated to reflect the current annotation of the sentence. The filename and the sentence number are listed at the top of these controls. Below that, two checkboxes reflect the label assigned to the sentence — *event sentence* or *nonevent sentence*. If both checkboxes are unchecked, the sentence is unlabeled. A label is assigned to the sentence by selecting one of the checkboxes. The interface prevents an annotator from selecting both checkboxes for a sentence. Sentences that have a label assigned to them, appear with a green background in the middle text area. This color-coding of sentences enables annotators to jump directly to the unannotated sentences and complete those annotations, thus speeding up the overall annotation process.

As mentioned in the guidelines, the event sentences in the annotations have an additional component — an *event identifier*. The event identifier helps segregate event sentences belonging to separate event descriptions in the document. This event identifier is selected from the list of identifiers appearing directly below the label checkboxes on the right side

of the GUI. An event sentence is only considered to be completely annotated (i.e., with a green background) when an event identifier from this list is selected along with the event sentence checkbox. The list of event identifiers available for selection is based on the IE answer keys specified during the "Setup Annotations."

The IE answer keys specified during setup are used to generate a summary of the events mentioned in the document. A unique integer corresponding to each event summary is used as the event identifier for the event. Thus, while annotating a sentence as an event sentence, the annotator selects one or more event identifiers from this list to complete the annotation of the event sentence. The set of event identifiers and summaries for a document is automatically generated from the IE answer key template associated with the document. Figure 5.4 shows example event identifiers and event summaries for a document containing two terrorist event descriptions. The two event identifiers correspond to the two answer key templates associated with the document. The event summaries paired with the identifiers (1) and (2) are generated from the templates by connecting some of the key pieces of information about the event listed in the template (*event type*, *target/victim* and *location*) with appropriate prepositions. In this example, the annotators would select either event

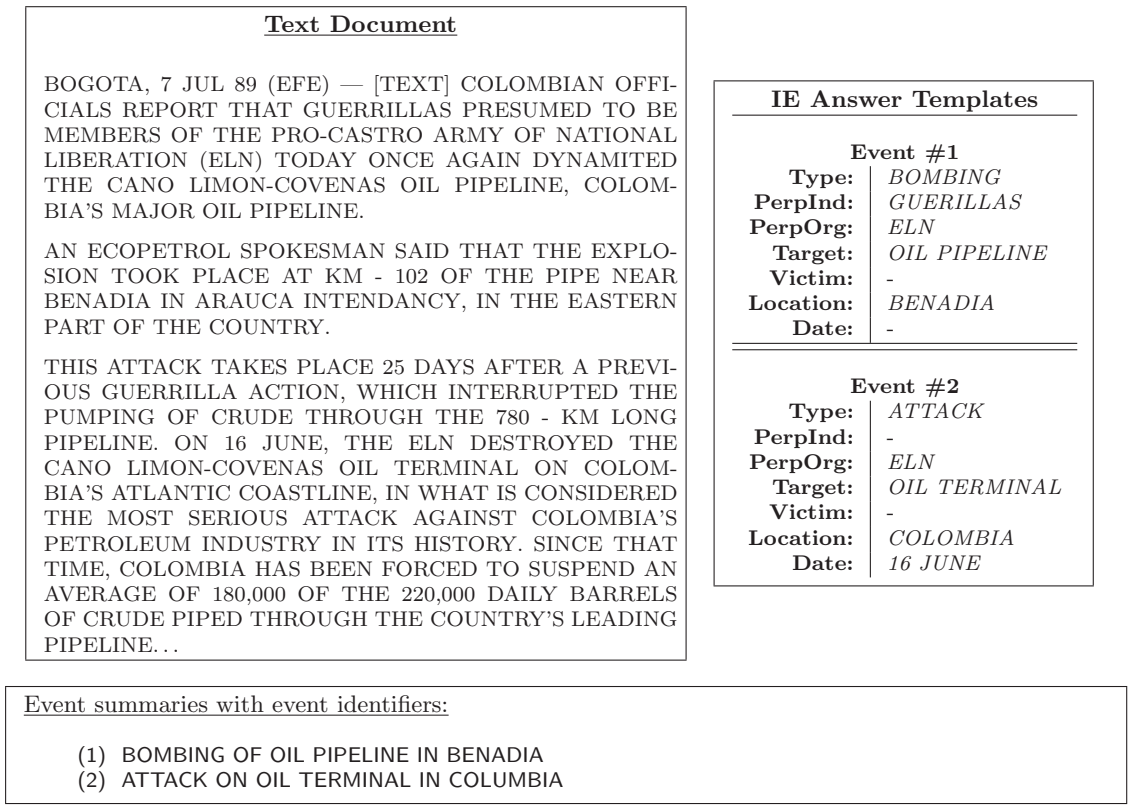


Figure 5.4: Example event summaries from IE event templates

identifier (1) or (2) or both for any sentence from the document that they label as an event sentence.

The interface described here essentially provides a front-end for the sentence annotations stored in text files in the output directory. The format of these annotation files is such that they could be potentially edited by other external tools (e.g., a text editor) in the absence of the annotation interface. Additionally, the format allows for increasing the scope and use of the annotations in future research. Figure 5.5 shows sample annotations on a disease outbreak document. Each sentence annotation consists of a sentence on one line of the text file followed by its labels on the following line. The sentence is indicated by the string "Sentence:" preceding the sentence. The label for that sentence appears on the following line within parentheses, preceded by the string "LabelSet." To allow for more fine-grained annotations in the future, the format uses EVENT_INCIDENT as the label for event sentences, and NOT_RELEVANT as the nonevent sentence label. Thus, in the future, additional labels from the guidelines, such as EVENT_CONSEQUENCE, EVENT_PRECURSOR and INCIDENTAL_INFORMATION can be incorporated. An earlier version of the interface allowed the annotators to assign a confidence score (integer ranging from 1 to 3) to the sentence labels. These confidence scores are stored by the format as an integer in square brackets following the label. Since the current version of the interface does not allow confidence scores, all labels get assigned a score of 3 by default. The format, therefore, allows for confidence scores to be potentially reintroduced in the future. Finally, the

```
Sentence: Indira Gandhi Memorial Hospital has recorded a sudden increase in patients admitted with
diarrhea accompanied by vomiting.
LabelSet(EVENT_INCIDENT[3]) EventID(1)

Sentence: The Department of Public Health said that viral hepatitis cases have also marked an increase
along with diarrhea patients.
LabelSet(EVENT_INCIDENT[3]) EventID(1,2)

Sentence: DPH quoted doctors as claiming that the cause for the cases were the consumption of
non-fresh food and impure water.
LabelSet(NOT_RELEVANT[3]) EventID()

Sentence: IGMH director Ali Mohamed told Haveeru that diarrhea patients marked an increase [since]
14 Feb 2000.
LabelSet(EVENT_INCIDENT[3]) EventID(1)

Sentence: Additional information about the etiology of this outbreak would be appreciated.
LabelSet(NOT_RELEVANT[3]) EventID()
```

Figure 5.5: File format of event sentence annotations

event identifiers associated with event sentences appear as comma separated integers in parentheses indicated by the string "EventID." Note that there are no event identifiers associated with nonevent sentences.

This completes the description of the interface, which affords great flexibility to the annotators. The *Save Annotations* option in the *Annotation* menu enables an annotator to save partially or fully completed annotations, close the interface, and come back to view or complete the annotations later. The architecture of the interface allows an annotator to quickly select and annotate any sentence in any document in the collection. Additionally, the annotation guidelines are accessible for quick reference through the *Help* menu. Overall, the point-and-click annotation interface facilitates relatively quick and convenient annotations of event/nonevent sentences in text documents.

5.2.3 Agreement Studies

The annotation guidelines and the annotation interface developed to manually assign labels to sentences are used to conduct an annotation study for this task. The primary goal of this study is to determine if it is possible for humans to perform this task adequately. The difficulty of this task for humans can give us an upper bound on the performance that can be expected from an automated system for this task. Another goal of this study is to investigate the extent to which people differ in their notion of event sentences. What it means for a sentence to be an event sentence can vary a lot from one person to the next. Therefore, an important question we would like to answer is: *given the guidelines for event sentence annotations, can humans agree on event sentences?* This annotation study explores these issues and presents a summary of its findings.

The annotation study starts with a training phase, where the annotators learn the annotation scheme and get familiar with the annotation interface. Part of the training phase was also used to refine the annotation guidelines to formulate a precise conceptual definition for event and nonevent sentences. The training phase was followed by an interannotator agreement study to measure the agreement of human annotators on identifying event sentences in text, based on the annotation guidelines provided. Finally, the annotators create gold standard data sets for two domains that are used to evaluate the performance of the sentential components of the Piper and Glacier IE models.

5.2.3.1 Training phase. One of the objectives of the training phase is to calibrate each annotator's notion of event sentences with respect to the specific provisions in the annotation guidelines. The guidelines essentially describe characteristics of sentences that make them event or nonevent sentences. These are high-level characteristics, *not tied to any specific type*

of event, based on the sentence types observed in Section 5.1. The training phase provides concrete annotation examples for specific types of events, and includes training exercises for annotators to understand how the guidelines are practically applied to real-world examples.

The training session started with a four-hour instructional session, which provided an overview of the annotation guidelines and examples of specific types of events. This session included paper-and-pencil exercises, where the annotators hand labeled sentences provided to them on paper. A similar paper-and-pencil training approach has been used in other annotation tasks, such as the annotation study conducted by Wiebe et al. [121] for sentiment and opinion annotations. Here we followed a similar training approach for event sentence annotation. The paper-and-pencil event sentence annotations were compared and verbally discussed among the annotators and the instructor, with a focus on the borderline cases and the reasons behind the annotations. The annotators then got a chance to practice annotations with the annotation GUI on several practice documents. These practice annotations were performed by the annotators on the laboratory PCs or home computer systems at their leisure. Just like the paper-and-pencil annotations, these practice annotations were also then compared and discussed in subsequent instructional training sessions. Several such practice annotations and instruction sessions were conducted, totaling about 22 hours of training per annotator. This included five in-class training sessions and five "at-home" exercises with the annotation GUI. In addition, the initial training sessions were also used to iteratively refine the annotation guidelines based on annotator feedback.

5.2.3.2 Interannotator agreement study. Two annotators, who were trained according to the training phase described above, participated in an interannotator agreement study to measure human agreement on this task. The annotators were computer science graduate students not involved in this information extraction project. They had not previously come across the data used for the study, and had no prior experience with such annotation. The annotators did have prior experience in Natural Language Processing, and thus were aware of the difficulty of the task.

The annotators were provided with 30 documents each for two types of events — *terrorist events* and *disease outbreak events*. These documents were randomly selected from the MUC-4 terrorist event data [114] and the ProMed disease outbreaks data [86, 82], respectively. These data sets are standard IE data sets (described earlier, in Chapter 2) that consist of documents containing event descriptions and corresponding IE answer key templates.[2] The annotators were asked to independently annotate event sentences in the

[2]Some documents in these data sets may contain no event descriptions. For this study, however, only

randomly selected documents using the annotation GUI.

Table 5.1 briefly summarizes some characteristics of the data, and some characteristics observed from the annotations. The 30 documents in the terrorist event data contain 454 sentences for annotation, making the average size of a document in this domain about 15 sentences long. On the other hand, the disease outbreaks data have 701 sentences for annotation, making the documents in this domain significantly larger, with an average size of about 23 sentences per document. All of these sentences are annotated by the two annotators (identified here as **Annotator #1** and **Annotator #2**), and Table 5.1 gives a breakdown of the event and nonevent sentences annotated by them on the two domains. The numbers show that **Annotator #2** tends to annotate more sentences as event sentences than **Annotator #1**. Across the two domains, **Annotator #1** consistently has between 6.3% and 7.0% fewer event sentences, which reflects the internal human perceptions of event descriptions.

Despite the difference in the number of event/nonevent sentence annotations, we see from Table 5.2a that the annotators achieve relatively good agreements on the two domains. The agreements are calculated as Cohen κ coefficients [21], which essentially compute the probability of the observed agreements over agreements by chance. The κ coefficient is given by the formula:

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{5.1}$$

where, P_o is the probability of the observed agreements, and P_e is the probability of agreements by chance. The annotators achieve an agreement of 0.77 on the terrorist events data and 0.72 on the disease outbreaks data, which according to the Landis and Koch [61] guidelines[3] for interpreting the Cohen κ represents "substantial agreement." In raw percent agreement, annotators achieve agreements of 90.3% and 88.7% on the two domains,

Table 5.1: Data characteristics and annotator characteristics

	Terrorist Events				*Disease Outbreaks*			
Data Files	30				30			
Total Sentences	454				701			
	Annotator #1		**Annotator #2**		**Annotator #1**		**Annotator #2**	
Event Sentences	121	(26.7%)	153	(33.7%)	166	(23.7%)	217	(31.0%)
Nonevent Sentences	333	(73.3%)	301	(66.3%)	535	(76.3%)	484	(69.0%)

documents with at least one event description were chosen for annotation.

[3]While these guidelines have not been completely substantiated, they do provide some insight into the extent of interannotator agreement for a given task.

Table 5.2: Results of the agreement study

(a) Interannotator agreement

	Cohen's κ	**95% Confidence Interval**	**Percent Agreement**
Terrorist Events	0.77	[0.68, 0.86]	90.3%
Disease Outbreaks	0.72	[0.65, 0.79]	88.7%

(b) Confusion Matrices

		Terrorist Events		*Disease Outbreaks*	
		Annotator #2			
		Event Sentences	Nonevent Sentences	Event Sentences	Nonevent Sentences
Annotator #1	Event Sentences	115	6	152	14
	Nonevent Sentences	38	295	65	470

respectively. Thus, this annotation study shows that there is relatively good agreement among human annotators for this sentence annotation task, within the specified definition for event descriptions.

Table 5.2b presents the confusion matrices for the annotation study on the two domains, displaying the raw numbers of agreements and disagreements between the annotators. The matrices show that the maximum number of disagreements occur in the cases where **Annotator #1** labels a sentence as nonevent, while **Annotator #2** labels it as an event sentence. Analyzing these specific examples indicates that many of the disagreements are on sentences that contain a reference to an event of interest, but there is some ambiguity regarding the mention of an event-specific detail in reference to the event. For example, in the following sentence:

> *This murder has made evident the lack of guarantees in the country, guarantees that the Government has systematically denied the opposition parties.*

the event "this murder" is mentioned, but there is disagreement about whether the sentence contains an event-specific detail. Another reason for many of the disagreements stem from an inconsistent notion of the event time frame between the annotators. Thus, incidents occurring at the edges of the time frame sometime have different labels from the annotators. These disagreements suggests that an automated system designed for this task likely will face similar issues in locating event sentences.

5.2.3.3 Gold standard data annotation. After completing the agreement study, and obtaining good human agreement scores on this task, we then created a gold standard

data set of event sentence annotations. A data set with human expert annotations can be used to evaluate the performance of the sentential components of IE models, and can help detect deficiencies in that component of the models. Additionally, such data can serve as the "perfect" sentential component to enable us to study the upper bounds on performance achievable in our IE models.

For the gold standard data set, our goal was to get human annotations for 100 documents (TST3) from the test set for terrorist events, and the entire 120 document test set for disease outbreaks. Of the 100 terrorist event documents, 31 documents have no relevant event described in them. All the sentences in these 31 documents are nonevent sentences. We, therefore, need human annotations for only 69 documents in this domain. Similarly, for the disease outbreaks data, we need human annotations for only 99 of the 120 documents (the remaining 21 contain only nonevent sentences). Furthermore, the 30 terrorism documents and 30 disease outbreaks documents which were used earlier in the agreement study were a subset of the 69 documents and 99 documents, respectively. Thus, we have human annotations on these from the agreement study. However, since we have two sets of annotations (from the two annotators) on these 30 documents in each domain, a process of "adjudication" was carried out to convert this into a single set of annotations. In this process, the two annotators came together and discussed the conflicting cases to come to a decision on the labels for these. Finally, the annotators annotated the remaining 39 terrorism documents and 69 disease outbreaks documents to complete the annotation of the gold standard data set. The entire undertaking, including the training phase, the agreement study and the creation of the gold standard data set, took a total of 77 person-hours of effort by the annotators.

5.3 Annotations with IE Answer Keys

Because of the time, expense and effort associated with developing human annotations, Chapter 3 and Chapter 4 both propose approximating these sentence-level annotations using IE answer key templates. Since such templates are available for a number of IE data sets, this method can be used to quickly generate approximate sentence-level annotations for those data sets. The annotations are then used in the pipelined Piper model and the unified probabilistic Glacier model for training the sentential components.

These approximate annotations from IE answer key templates are done by "reverse-engineering" the answer keys with the corresponding documents. Given a text document and the corresponding answer key template, the system locates within the document each

of the answer key strings listed in the template. Any sentence in the document found to contain one of these answer key strings is then labeled as an event sentence, and the remaining sentences are labeled as nonevent sentences. The basic assumption underlying this approximate annotation scheme is that since the answer key templates represent the information from the event descriptions, any sentence containing this information (the answer key strings), most likely, is part of an event description.

However, as mentioned in Chapter 3 and in Chapter 4, this assumption does not always hold. Many times, an answer key string appears at multiple places in the document, only some of which are part of an event description. As a result, the annotations obtained from this process can be noisy. The questions that then come up are: *How noisy are these annotations? Can we do away with the human annotations altogether?* Now that we have done a human agreement study and have a gold standard data set created by human experts, we are in a position to answer these questions.

One way to evaluate the approximate annotations from IE answer key templates is to consider this approximate annotation technique as just another annotator in an agreement study. To do this, approximate annotations are generated for the 30 documents (for each domain) used in the human agreement study described earlier. Pretending that this annotation technique is **Annotator #3** in the agreement study, we can now compare its annotations to those of the human annotators.

Table 5.3 briefly summarizes some characteristics of the data, and some annotator characteristics observed from the annotations. Note that these annotations behave quite differently in the two domains. For terrorist events, the number of event sentences identified by the answer keys is in the same ballpark as those identified by the two human annotators — 31.7% event sentences from the approximate annotations, compared to 26.7% and 33.7% identified by the two human annotators, respectively. For disease outbreaks, however, the

Table 5.3: Approximate annotation characteristics

Terrorist Events						
Data Files	30					
Total Sentences	454					
	Annotator #1		**Annotator #2**		**Annotator #3**	
Event Sentences	121	(26.7%)	153	(33.7%)	144	(31.7%)
Nonevent Sentences	333	(73.3%)	301	(66.3%)	310	(68.3%)
Disease Outbreaks						
Data Files	30					
Total Sentences	701					
	Annotator #1		**Annotator #2**		**Annotator #3**	
Event Sentences	166	(23.7%)	217	(31.0%)	308	(43.9%)
Nonevent Sentences	535	(76.3%)	484	(69.0%)	393	(56.1%)

number of event sentences is quite a bit (over 12%) more than those identified by the human annotators. The approximate annotations identify almost 43.9% of the sentences as event sentences, compared to 31.0% and 23.7% for humans.

This difference in the numbers of event/nonevent sentences across the two domains is reflected in the agreement scores for the approximate annotations. Table 5.4 contains the agreement scores in this setup (i.e., considering the approximate annotations as **Annotator #3**). The pairwise Cohen κ is calculated for each pair of annotators (#1 and #2, #1 and #3, #2 and #3), and the average of this pairwise κ is presented in the table. The average pairwise Cohen κ coefficient in this interannotator agreement study is 0.70 for terrorist events, which is 0.07 lower than the human agreements. For disease outbreaks, however, the average Cohen κ is 0.43, almost 0.29 lower than the human agreements! This difference in agreement scores with the approximate annotations indicates that approximate annotations with answer keys can be close to human judgement for certain types of events, but may not consistently apply across all event types. This difference can also be outcome of the difference in the document characteristics across the two domains.

Using answer key templates to locate event sentences makes the assumption that any mention of an entity listed in an answer key must appear only in event descriptions in the text document. This assumption does not always hold. An entity may be mentioned multiple times within a document, not all of which may be within event descriptions. From Table 5.3 and Table 5.4 we see that the assumption seems to hold, for the most part, in terrorist event descriptions, but breaks down in the case of disease outbreaks. Reading through the disease outbreaks data we find that diseases are mentioned many times in the documents, but many of these disease mentions are in nonevent contexts. Typically, once the disease outbreak has been mentioned, other general characteristics of the disease are described. These nonevent sentences get tagged as event sentences in the approximate annotation scheme.

5.4 Event Role Analysis

Now that we have explored the use of IE answer key templates for approximating event sentence annotations, let us take a look at another aspect of IE answer keys — their

Table 5.4: Interannotator agreement scores with approximate annotations

	Average Cohen's κ	**95% Confidence Interval**	**Avg. Percent Agreement**
Terrorist Events	0.70	[0.61, 0.80]	87.4%
Disease Outbreaks	0.43	[0.36, 0.50]	73.9%

relationship with event roles. Using IE answer key templates for identifying event sentences relies on the assumption that entities playing a specific role in an event are likely to appear only within the description of an event of interest. Thus, an implicit relationship between event roles and event sentences is assumed. This section explores this relationship and other characteristics of the data in greater depth.

Recall that the pipelined model for IE (PIPER) described in Chapter 3 first locates event sentences in text, and then uses a text extraction component to identify the role fillers within these sentences. The question that arise here are: *Do all the role fillers appear within the event sentences? Or are we likely to lose some of these, which happen to appear in nonevent sentences?* Essentially, what we would like to measure is the *maximum achievable recall* as a result of discarding nonevent sentences. When we have all the sentences to extract information from, the maximum achievable recall is 1.0, since all the role fillers are present within these sentences. However, once we discard some of the sentences, some event roles may get discarded in the process. This situation can occur when a role filler is not mentioned as part of an event incident, but in a non event sentence containing incidental information or an event consequence. For example, consider the sentences:

> *Three armed men attacked a military convoy in Iraq yesterday. The attackers are believed to be members of Al Qaeda.*

The first sentence describes a terrorist event, while the second sentence provides incidental information about the attackers mentioned in the event description. According to the annotation guidelines, the second sentence is a nonevent sentence. However, it does contain information about the organization that the perpetrators belong to. Thus, we would like to measure the effect of discarding nonevent sentences on such event roles.

We conducted an experiment to measure the maximum achievable recall on the human annotated event sentences (gold standard data). The idea was to see if having perfect event sentence annotations could enable us to locate all the role fillers for an event. We simulated a perfect extraction system that extracts event role fillers only from the event sentences. For terrorist events, we find that the perfect extraction system can achieve a maximum possible recall of 0.89 if applied only to event sentences. This means that 11% of terrorist role fillers are in the nonevent sentences. For disease outbreaks, the maximum achievable recall is 0.94 within event sentences. Table 5.5 shows the breakdown of this maximum achievable recall for the various event roles in the two domains. The "Perpetrator Individual" and "Perpetrator Organization" event roles in the terrorism domain are the two that occur most often in nonevent sentences. Over 20% of these entities are mentioned only within nonevent

Table 5.5: Maximum achievable recall within event sentences

Terrorist Events	
Perpetrator Individual	0.78
Perpetrator Organization	0.77
Physical Target	1.00
Human Victim	0.93
Weapon	1.00
Disease Outbreaks	
Disease	0.95
Victim	0.94

contexts. This observation illustrates the need for our probabilistic model (GLACIER) over the pipelined model (PIPER), which discards nonevent sentences.

5.4.1 Impact of Event Recognition on Recall

The observation about the maximum achievable recall illustrates the potential loss of recall one might face with the use of event/nonevent sentence information in IE. However, there are also benefits of incorporating event sentence information in IE, which we highlight here. We find that the two main characteristics of data that affect the performance of an IE system are *redundancy of answer opportunities* and *density of event sentences*. Redundancy of answer opportunities is the number of relevant answer strings in the text corresponding to each event role in the IE answer key. This primarily affects the recall of an IE system. Greater redundancy provides multiple opportunities for an event role extraction system to locate the correct extractions. For example, *elephants* that are victims of a disease outbreak may be mentioned multiple times in a document. Extracting any one of those mentions results in a correct extraction for that event role. Light et al. [65] have shown, in the context of Question Answering, a direct correlation between redundancy in answer opportunities and system performance. Density of event sentences is the proportion of event sentences in the data set. It directly affects the precision of an IE system. Each nonevent sentence in the data translates to a greater chance of incorrect extractions from the nonevent sentences.

To better understand the extent of the redundancy of answer opportunities and their variation across event roles and domains, we measured the redundancy of answer opportunities on our two domains — terrorist events and disease outbreaks. We also measured the effect of classifier performance trade-offs on this redundancy, by randomly discarding some of the sentences containing event role fillers. Figure 5.6 shows the findings from this experiment for the two data sets (on 200 documents for terrorist events and 120 documents for disease outbreaks). The x-axis represents the percentage of sentences discarded, and the

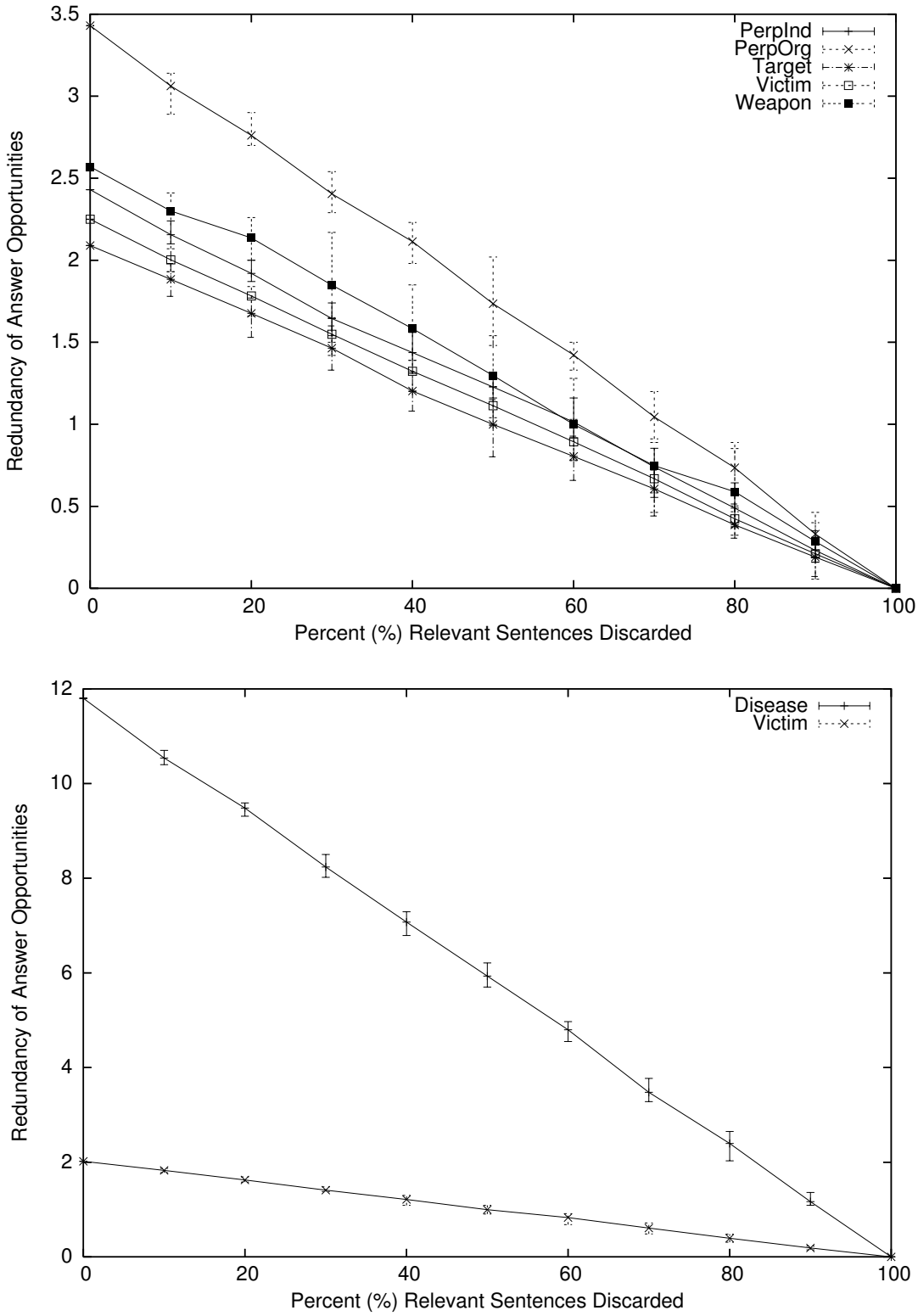


Figure 5.6: Effects of classifier performance on redundancy

y-axis plots the average number of answer opportunities in the text for each answer in the answer key templates. The error-bars indicate the variation in these values across ten runs.

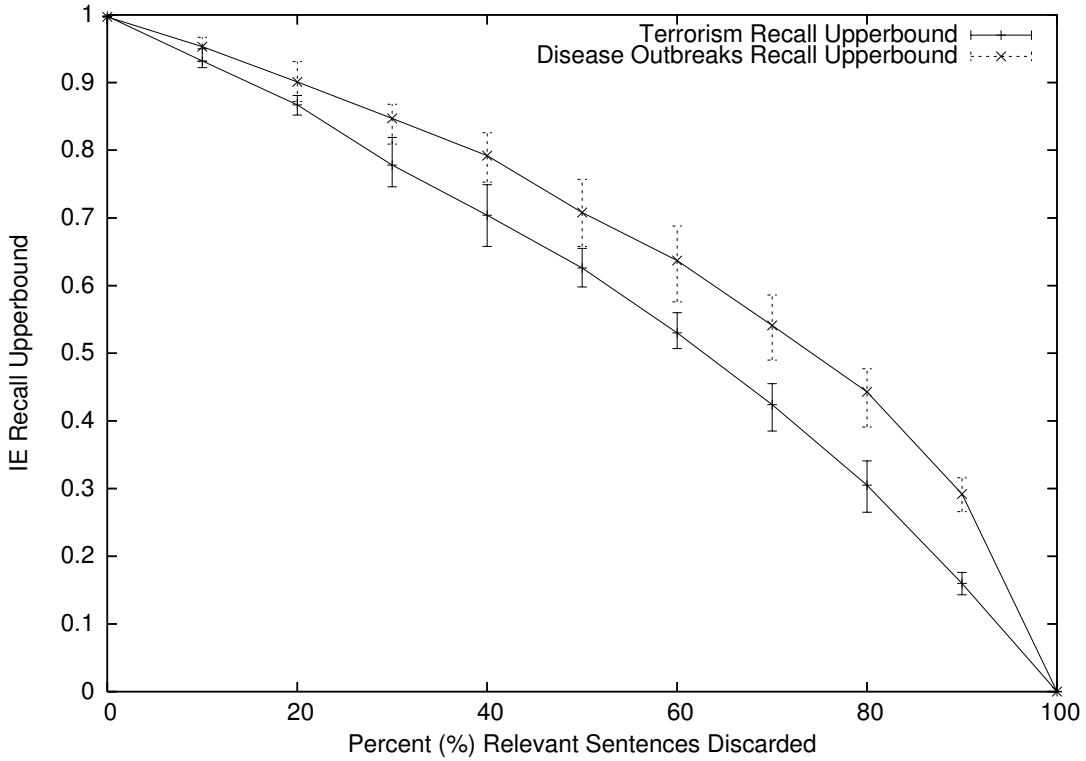
Overall, the average amount of redundancy is about 2.5 answer opportunities for each terrorism event answer, and nearly 6 answer opportunities in the disease outbreaks data set. However, the more interesting observation is that the different event roles have different amounts of redundancy. In the terrorist event data set, the *perpetrator organization* answer opportunities appear almost 3.5 times per answer in the data set. In the disease outbreaks data, the *disease* role filler strings appear almost 12 times per answer! Even if 50% of the sentences were discarded, we would be left with about 6 disease answer opportunities per answer.

In addition, we see that the amount of redundancy decreases linearly as we randomly discard sentences. After discarding about 40% of the sentences containing event roles in the two data sets we still have about 1.5 answer strings for each correct answer. Thus, some weaknesses in event recognition can be tolerated by the IE system. A reasonable extraction coverage could potentially be achieved even with a sentential event recognizer that incorrectly discards many event sentences.

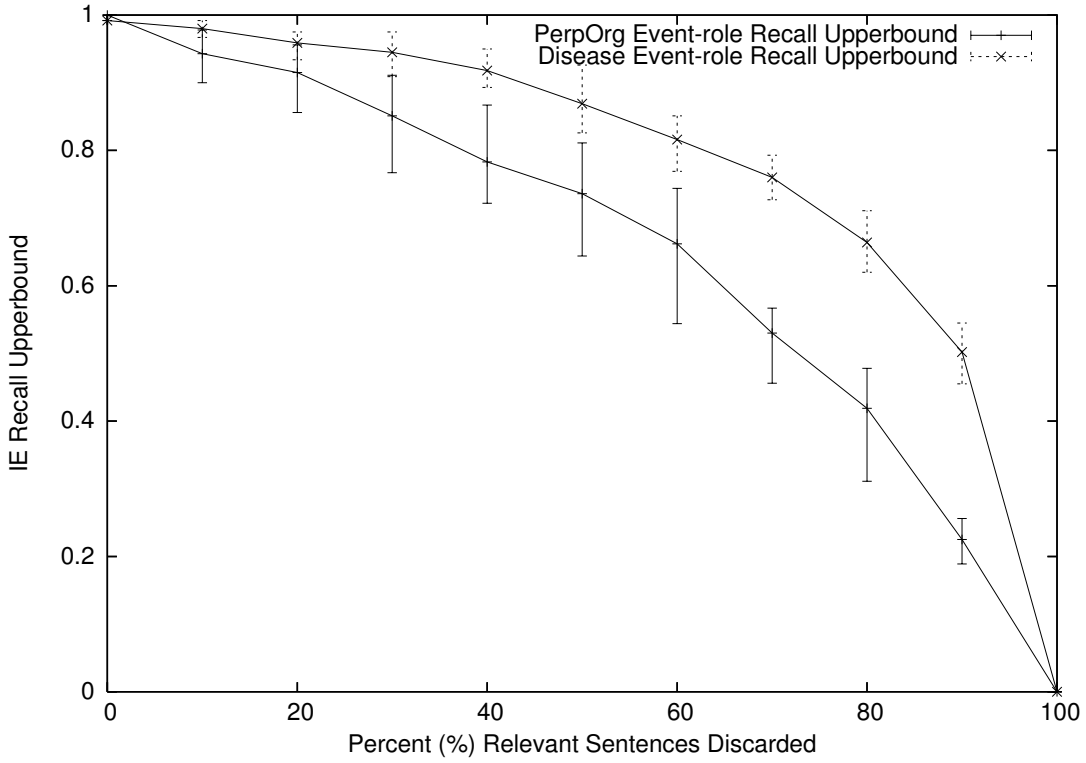
To test this hypothesis we ran a second experiment. This experiment measures the effect on *maximum achievable recall* of randomly discarding sentences containing role fillers in each data set. Figure 5.7a plots the maximum achievable recall on the y-axis corresponding to the percentage of discarded sentences on the x-axis (in the two data sets). The effect of the redundancy is that the maximum achievable recall drops slowly initially, and the drop is sharper as the percentage of discarded sentences approaches 100%. What this means is that even after losing 30% of the sentences containing role fillers, an IE system could still potentially achieve a recall of 0.78 on the terrorist events data, and 0.85 on the disease outbreaks data. As seen in Figure 5.7b, this effect is far more pronounced for the individual event roles with high redundancy — *perpetrator organization* in the terrorism data set and *disease* in the disease outbreaks data set.

5.4.2 Impact of Event Recognition on Precision

One of the goals of this research is to design classifiers for *sentential event recognition* as a component in the two IE models. We hope that such classifiers would eliminate only those sentences that do not contain event role fillers, and keep those sentences that contain an event role filler. The extraction side of these models can benefit from such a classifier by having fewer false-positive extractions in their search for role fillers within the event-specific regions of the text. The sentential event recognizer acts as a filter in this model, and discards



(a) Maximum achievable recall in the two data sets



(b) Maximum achievable recall for PerpOrg and Disease

Figure 5.7: Effects of classifier performance on maximum achievable recall

nonevent sentences. This in turn can improve the precision of an extraction system, by preventing incorrect extractions from nonevent sentences.

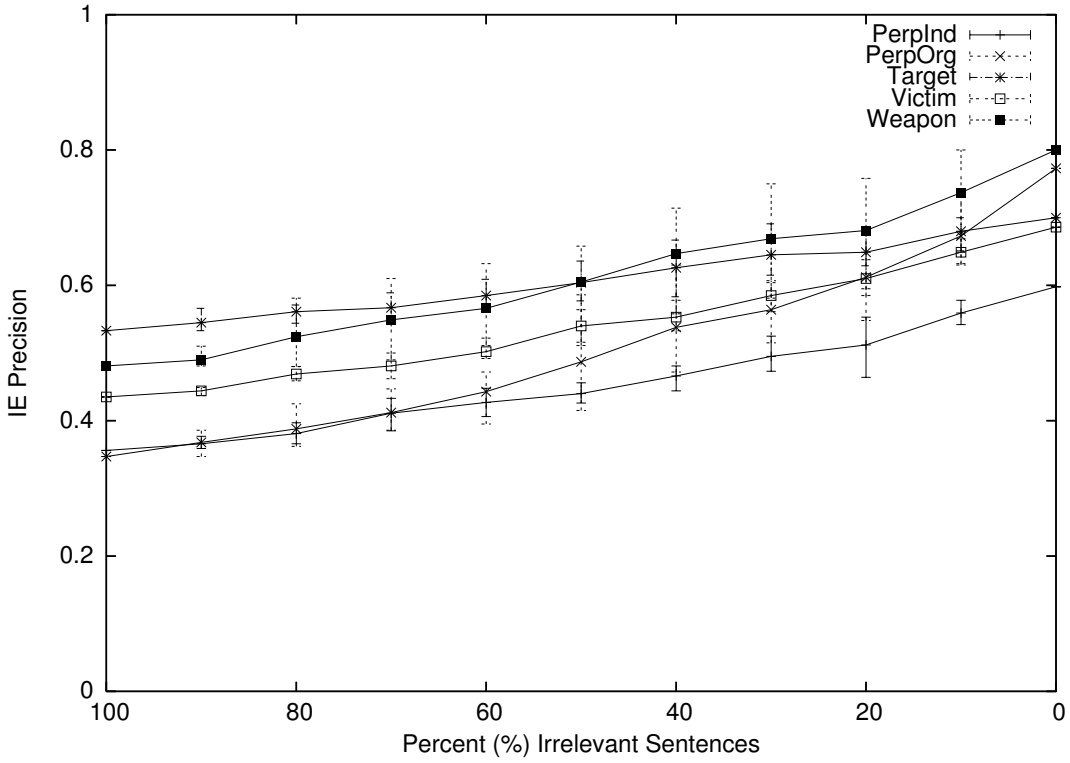
To determine the extent to which an IE system could possibly benefit from event sentence recognition, we devised an experiment to empirically analyze the effect of such sentences on IE precision. We started by measuring the proportion of sentences containing event role fillers for each data set and found that only 19% of the sentences in the terrorism data set contain role filler strings, and about 28% of the sentences in the disease outbreaks data contain role filler strings. The large number of sentences containing no event role fillers can cause an IE system to incorrectly extract information from nonevent sentences, negatively affecting its precision. We artificially filter these data by randomly discarding from the two data sets those sentences that contain no role fillers. We apply an existing pattern-based IE system to these data and measure its precision. For this experiment, we used the AutoSlog-TS system [92, 95], which is a weakly supervised pattern-based IE system. It comes trained for the terrorist events and the disease outbreaks IE tasks.

Figure 5.8 shows the effect of varying amounts of irrelevant sentences on the precision of the IE system in two domains. On each graph, the x-axis plots the percentage of these irrelevant sentences in the data set, while the y-axis plots the precision of the AutoSlog-TS IE system applied to this data set. We observe that the precision of the IE system increases almost linearly as the irrelevant sentences are discarded. Furthermore, we see a nearly 20% precision gain when the data contain only sentences with at least one role filler. Comparing the various event roles, we find that the irrelevant sentences have a much greater effect on event roles such as *weapon* and *perpetrator organization* in the terrorism data set. Since most role filler strings tend to appear in event sentences, this experiment demonstrates the potential benefits of sentential event recognizers, to eliminate as many of the nonevent sentences as possible.

5.5 Summary of the Analyses

Various types of analyses of data have been presented in the preceding sections, providing insights into the nature of text we are dealing with. Before concluding this chapter, we briefly summarize the key observations from these analyses:

1. In developing the annotation guidelines for the interannotator agreement study, we identified five types of sentences, based on their purpose within event descriptions. These sentence types were useful in establishing a clear definition for event sentences in text.



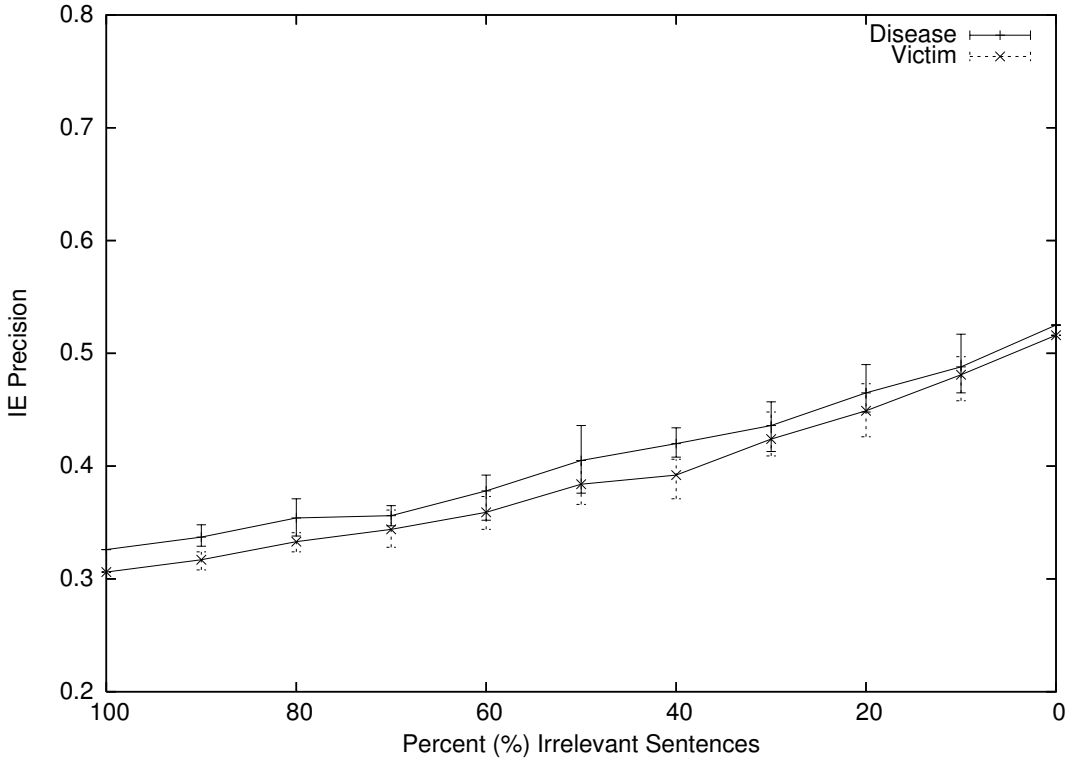


Figure 5.8: Effect of nonevent sentences on IE precision

2. Based on the annotation guidelines developed, an annotation study was conducted to determine if humans agree on the notion of event sentences. The study finds that there is strong agreement between humans, when given a specific definition of event sentences. An interannotator agreement of 0.72 Cohen κ and 0.77 Cohen κ is attained by annotators on two event types.

3. Having conducted the interannotator agreement study, an experiment was conducted to determine if the approximate annotations using IE answer key templates match up with human annotations. The experiment shows that the answer to this varies by different event types. For terrorist event descriptions, a relatively good agreement is seen between the approximate annotations and the human annotations. However, for disease outbreak events, there is very little agreement.

4. An experiment was conducted to determine if all the event role fillers can be found within event sentences. The results show that for terrorist events 89% of the event role fillers are mentioned in the event sentences, and for disease outbreaks this value is 94%. This suggests that some event role fillers are mentioned only within nonevent sentences, and would likely be missed by our pipelined model. The experiment shows that the *perpetrator organization* and *perpetrator individual* in terrorist events tend to be mentioned in nonevent sentences about 22% of the time.

5. The redundancy of an event role is the number of answer strings corresponding to one event role filler. An event role with high redundancy can achieve good recall even if some of the answer strings are discarded by the sentential event recognizer. An experiment was conducted to determine the "tolerance" of the data sets and the various event roles to such discarded strings. The experiment shows that the terrorist events data have a redundancy of about 2.5, and the disease outbreaks data have a redundancy of about 6 answer strings per event role filler.

6. The large proportion of nonevent sentences in the data set makes the task of finding event role fillers extremely challenging. An experiment was conducted to predict the kinds of performance improvements we could hope to see if all of the nonevent sentences are discarded. This effects the precision of IE systems, and our experiment finds that an existing pattern-based achieves a 20% improvement in precision by correctly discarding nonevent sentences. The potential improvement can be even more if the IE system is specifically designed to deal only with event sentences.

The various types of data analyses indicate that recognizing event sentences before performing event role extraction can be beneficial for IE, and should benefit our PIPER and GLACIER models. The design of the sentential event recognizer can be tailored to the specific characteristics of the data to extract maximum benefit from the system.

CHAPTER 6

EMPIRICAL EVALUATION

This dissertation presents two novel approaches for IE that combine information from the wider sentential context with the local contextual evidence to make inferences about event role extractions. The first approach is a pipelined model (PIPER) applying a sentential event recognizer to text documents, followed by a pattern-based extraction system for identifying event roles within event sentences. Addressing the drawbacks of this pipelined approach is a unified probabilistic model for IE (GLACIER), which avoids making discrete decisions by using probabilities to gently balance the influence of the two components. In this chapter, we evaluate the performance of these models. Section 6.1 lays down the plan for this evaluation. Section 6.2 first evaluates the sentential components of the IE models. Section 6.3 describes the data and methodology used in the IE evaluation. Section 6.4 then describes the baseline IE systems that are compared with our new models, and presents IE evaluation results for these baselines. This is followed in Section 6.5 and Section 6.6 by an extensive evaluation of the PIPER and GLACIER models for IE. An analysis of the features used is presented in Section 6.9, and a summary of the evaluation and our observations therein are finally presented in Section 6.12.

6.1 Evaluation Plan

The goal of this research is to perform effective extraction of event role information from free text. Our evaluation of the IE models presented in this dissertation is based on their performance at the task of identifying event role fillers of relevant events. Implementations of the IE models are applied to text documents to extract event role fillers, and the extracted information is compared against gold standard IE event templates to estimate the IE accuracy. Note that this differs from the complete IE task (the *Scenario Template* task), which requires the IE system to create event templates — one template per event. Our work focuses on extracting event role fillers and not on template generation per se (e.g., we are not concerned with coreference resolution or which extraction belongs in which template). Consequently, our experiments evaluate the accuracy of the extractions individually.

In this dissertation, all of the evaluation is done for two event types — *terrorist events* and *disease outbreaks*. Each of these event types has it corresponding data sets containing documents, along with the answer key templates. The use of two data sets for the evaluation aims to illustrate the portability of the models for various IE tasks, and also serves to bring forth various issues associated with cross-domain portability. The terrorist events data set is a standard IE data set, which was developed for a comparative evaluation of IE systems participating in the MUC-3 [113] and MUC-4 [114] conferences. The disease outbreaks data set [86, 82] was created from reports of disease outbreaks posted to ProMed-mail, a freely accessible global electronic reporting system for outbreaks of infectious diseases. A more detailed description of both these data sets appears in Chapter 2 (Section 2.6).

Since the models for IE presented in this dissertation all rely on the detection of event descriptions in text, we would first like to measure the efficacy of this individual component. The weaknesses of individual components can contribute to the accuracy of the overall IE system, and the evaluation of this component can indicate the contribution of the sentential event recognizer to the IE model. Before performing the complete IE evaluation, therefore, an evaluation of the sentential event recognizers in the IE models is first presented. The evaluation is performed on the two event types using the human annotated gold standard data created in Chapter 5 (Section 5.2).

Additionally, several variations of the sentential components of the IE models are tuned using tuning data in the two domains. The sentential event recognizers in the PIPER model identify event sentences in text using classifiers, whose parameters can be tuned. For example, the proportion of event sentences identified by a classifier can be influenced by choosing an appropriate threshold for classification. This is especially important for IE, because usually such data are highly skewed. The number of nonevent sentences far surpasses the number of event sentences for typical IE data sets. This data skew can affect the GLACIER model too. Thus, some tuning of these classifiers is needed.

Following the sentential component evaluation, a full-fledged evaluation of the IE models is performed. Implementations of the two IE models are applied to the test documents in the data sets, to extract event role fillers of the events described. The extracted role fillers are matched against the role fillers listed in the answer key templates associated with the documents. Based on this, the precision, recall and F-score of the extractions are used to measure the performance of the IE models. These performance metrics are used to compare our new IE models with one another and with other IE models that do not have the advantage of event sentence information. Thus, the following sections aim to demonstrate

the prowess of our IE models through empirical evaluation.

6.2 Sentential Event Recognizer Evaluation

Chapter 5 presented several analyses of the data indicating various potential benefits and risks associated with the use of sentential event recognizers in the IE systems. We now study the performance of our sentential event recognizers in practice. The sentential components of our two IE models (PIPER and GLACIER) are evaluated with respect to human annotations of sentences, to provide us with an estimate of their effectiveness at this subtask.

6.2.1 The PIPER Model

Chapter 3 (Section 3.2) described three classifier-based variations of the sentential event recognizer used in the PIPER model for IE. All three variations classify sentences as event/nonevent sentences based on features associated with the sentences. The differences are primarily in the strategies employed for training these components: (a) using IE answer key templates (**Anskey**), (b) self-training with seed patterns and document labels (**Self**), and (c) multiple instance learning with document labels (**MIL**). All three variations were evaluated against human annotations of event sentences on two types of events — terrorist events and disease outbreaks.

6.2.1.1 Anskey. An SVM classifier and a Naïve Bayes classifier were trained using approximate sentence annotations with IE answer key templates. The SVM^{light} [56] implementation of the SVM and the Weka Toolkit [122] implementation of the Naïve Bayes were used. The SVM employed a linear kernel with the default configuration from the software package, and the Naïve Bayes used its default configuration in the Weka Toolkit. For terrorist events, 1,300 MUC-4 documents with corresponding answer keys were used for training. This resulted in 3,092 positive training examples (sentences), and 16,221 negative training examples (sentences). The feature vectors representing the training examples used the eight types of features described in Chapter 3 (Section 3.3). These features were generated from the training data, and a frequency cutoff of four was applied to the features. Only features appearing four or more times in the training data were used in the feature vectors; the remaining features were discarded. This generated a feature set of 14,432 features in each feature vector. For disease outbreaks, the training data set was much smaller, containing 125 ProMed documents with answer keys. These provided 1,005 positive training examples and 2,546 negative training examples. Here too a frequency cutoff of four was applied to the feature set, resulting in 3,978 features in each feature vector.

Observe that the training data are highly skewed in favor of the negative examples. In the test data too, the number of negative examples far surpasses the number of positive examples. As a result of this skew, classifiers trained on these data tend to be biased towards the negative class. An SVM or a Naïve Bayes classifier trained on this data tends to achieve a high accuracy by classifying most examples as negative. Consequently, it turns out that the recall of such a classifier on identifying event sentences is relatively low. Applying the trained classifiers to the human-annotated test sentences (100 terrorist event documents and 120 disease outbreaks documents) we obtained the results shown in Table 6.1. The table lists the overall accuracy of the classifier, and its precision, recall and F-score on the two classes. Note that the recall for the SVM on event sentences is 0.41 and 0.31 on the two domains, respectively. This implies that almost 59% to 69% of the event sentences were being mislabeled as nonevent sentences by the classifier.

Because most of the event role fillers are mentioned within event sentences, a low recall on these sentences can be a problem for our IE models. A low recall on event sentences results in many event sentences getting incorrectly labeled as nonevent sentences by the classifier. As a result of this, an extraction model can fail to extract role fillers present in these sentences. Thus, the effect of data skew on sentential event recognition recall is an issue that needs to be addressed in our IE models.

One way in which the effect of skewed data can be countered is to try alternate thresholds of the decision boundaries used by the classifiers in making their decisions. In case of SVMs, the decision is made based on the value of the decision function applied to the test example. A test example is classified as positive if this value is greater than zero, negative otherwise. A Naïve Bayes classifier computes the probability of the two classes and assigns the class with the higher probability (which implicitly implies a threshold of 0.5 for a binary classifier). Thus, we try other thresholds to influence the balance of positive and negative classes identified by the classifiers.

To identify the best thresholds in our two classifiers, they were applied to the tuning

Table 6.1: Evaluation of Anskey classifiers

		Event			**Nonevent**		
	Acc	Pr	Rec	F	Pr	Rec	F
Terrorist Events (MUC-4)							
SVM	0.89	0.83	0.41	0.55	0.89	0.98	0.94
NB	0.83	0.50	0.70	0.58	0.94	0.86	0.90
Disease Outbreaks (ProMed)							
SVM	0.75	0.43	0.31	0.36	0.81	0.88	0.84
NB	0.74	0.45	0.53	0.48	0.85	0.80	0.82

documents in each domain, and evaluated against the answer key annotations (since human annotations are not available on the tuning data). The tuning data consist of 200 documents (TST1+TST2) for terrorist events and 20 documents for disease outbreaks, all with corresponding answer key templates. Table 6.2 shows the results of using various thresholds for the classifiers in the two domains. For the SVM, the decision values were normalized between 0.0 and 1.0, and several thresholds within this range were applied. The Naïve Bayes used probability values for its classification, which were already in between the 0.0 and 1.0 range. For this classifier too, several thresholds in this range were applied.

The table shows that the SVM achieves the highest accuracy at the 0.5 threshold. However, at this threshold the F-score of the event sentences is only 0.32 for terrorist events and 0.46 for disease outbreaks. At a threshold of 0.3, we observe a much higher F-score of 0.60 and 0.58 (and a higher recall) on event sentences for the two domains respectively. This increase in F-score is at the cost of a slightly lower accuracy and a slightly lower F-score on nonevent sentences. But this trade-off is acceptable if we want a higher recall on event sentences. The 0.3 threshold for the SVM corresponds to a decision function value of -0.82 for terrorist events and -0.57 for disease outbreaks. Thus, we chose these decision function values as thresholds for the SVM classifier. The Naïve Bayes classifier, on the other hand, was not affected very much by the different thresholds. So, we kept the default 0.5 threshold of the Naïve Bayes for labeling event sentences.

Having used the tuning set to select thresholds for the classifiers, the chosen classifiers were then applied to the human-annotated test sentences to gauge their true performance on identifying event sentences in text. Table 6.3 presents the results of applying these selected classifiers (with the chosen thresholds) to the 100 human-annotated documents for terrorist events, and the 120 documents for disease outbreaks. Compared to their unmodified counterparts in Table 6.1 the SVM accuracies are slightly lower, but their performance

Table 6.2: Identifying a threshold for classifiers

		Terrorist Events (MUC-4)							*Disease Outbreaks (ProMed)*						
			Event			**Nonevent**				**Event**			**Nonevent**		
		Acc	Pr	Rec	F	Pr	Rec	F	Acc	Pr	Rec	F	Pr	Rec	F
SVM	.1	0.22	0.20	0.99	0.33	0.94	0.03	0.07	0.45	0.39	0.95	0.56	0.85	0.16	0.26
	.3	0.83	0.55	0.66	0.60	0.91	0.87	0.89	0.69	0.58	0.58	0.58	0.76	0.76	0.76
	.5	0.84	0.86	0.20	0.32	0.84	0.99	0.91	0.72	0.76	0.33	0.46	0.71	0.94	0.81
	.7	0.81	0.95	0.03	0.06	0.81	1.00	0.90	0.68	1.00	0.13	0.23	0.66	1.00	0.80
	.9	0.81	1.00	0.00	0.01	0.81	1.00	0.89	0.65	1.00	0.03	0.05	0.64	1.00	0.78
NB	.1	0.82	0.52	0.68	0.59	0.92	0.85	0.88	0.71	0.60	0.66	0.63	0.79	0.74	0.77
	.3	0.82	0.53	0.65	0.59	0.91	0.87	0.89	0.72	0.62	0.60	0.61	0.77	0.78	0.78
	.5	0.83	0.55	0.63	0.58	0.91	0.88	0.89	0.72	0.63	0.56	0.59	0.76	0.81	0.79
	.7	0.83	0.56	0.60	0.58	0.90	0.89	0.89	0.72	0.64	0.55	0.59	0.76	0.82	0.79
	.9	0.84	0.58	0.57	0.57	0.90	0.90	0.90	0.70	0.63	0.47	0.53	0.73	0.84	0.78

Table 6.3: Evaluation of best Anskey classifiers on human annotations

		Event			Nonevent		
	Acc	Pr	Rec	F	Pr	Rec	F
Terrorist Events (MUC-4)							
SVM (−0.82)	0.86	0.54	0.75	0.63	0.95	0.88	0.91
NB (0.5)	0.83	0.50	0.70	0.58	0.94	0.86	0.90
Disease Outbreaks (ProMed)							
SVM (−0.57)	0.69	0.38	0.52	0.44	0.84	0.75	0.79
NB (0.5)	0.74	0.45	0.53	0.48	0.85	0.80	0.82

on identifying event sentences is markedly different — achieving a higher recall in both domains. The Naïve Bayes results are unchanged from before, since we use the default threshold (of 0.5) for this classifier.

6.2.1.2 Self. To reduce the amount of supervision used in training the classifiers, a self-training strategy was introduced to train the sentential event recognizer. The self-training strategy employed for training an SVM sentence classifier uses only a set of seed extraction patterns, a set of relevant documents and a set of irrelevant documents. Using these data the SVM is iteratively self-trained to obtain a classifier that is then used to identify event sentences in text. The details of this training and an evaluation of the classifier are presented here.

For terrorist events, the sets of relevant and irrelevant documents used in the self-training were obtained by separating the 1,300 MUC-4 documents into two sets — documents containing the description of a relevant event and those that do not contain any relevant events. This was determined by using the answer key templates associated with the documents. Documents that have a nonempty answer key template are relevant documents, while those that do not are not. This resulted in 700 relevant and 600 irrelevant documents for terrorist events. For the disease outbreaks domain we had only a small amount of data with answer key templates. Consequently, the set of relevant and irrelevant documents were collected in a different manner for this domain. The relevant document set was collected by downloading disease outbreak reports from ProMed-mail,[1] a freely accessible global electronic reporting system for outbreaks of diseases. Since the online resource is designed for the submission of disease outbreak reports, most of the documents obtained from here were relevant documents. The irrelevant documents were obtained from PubMed,[2] an online repository of biomedical journal abstracts. Specific queries were used to ensure that the

[1] http://www.promedmail.org

[2] http://www.pubmed.gov

documents collected from PubMed were not related to disease outbreaks. As a result of this process, we obtained 2,000 relevant and 4,000 irrelevant documents with roughly equal word counts for disease outbreaks.

While the seed patterns used to start off the self-training can be manually created by human experts, here they were semiautomatically generated from the relevant and irrelevant documents sets. First, an exhaustive set of patterns was automatically generated using the AutoSlog-TS [92] system. This set is exhaustive in the sense that patterns are generated to extract literally every noun phrase in the relevant and irrelevant documents. Appendix C describes the types of patterns generated by the system. Of these patterns, those appearing in the data set 50 times or more were ranked by their probability of appearing within a relevant document. The top 20 patterns from this ranked list (for each domain) were chosen as the set of seed patterns. We had several ties in probability values, and thus obtained more than 20 seeds in each data set. To reduce each seed set to 20, the last-ranked patterns were manually inspected by a human expert and the least relevant patterns were discarded. The resulting seed patterns are listed in Table 6.4.

The documents and the seed patterns were used in the iterative self-training process to generate an SVM for each domain. The process employed the SVM^{light} [56] implementation of an SVM, using the default configuration with a linear kernel. All of the features that were used by the Anskey SVM model described earlier were also used for this self-trained model — a total of 14,432 features and 3,978 features for the terrorist events and the disease outbreaks, respectively.

The self-training strategy implemented here does not define an automatic stopping criteria for the iterative process. Additionally, the SVM here had the same problem that was encountered with the Anskey version — that of skewed data. As a result of the skewed data, the SVM tends to err on the side of classifying sentences as nonevent sentences, resulting in a lower recall on event sentences. To remedy this, once again we looked at thresholds on the decision function of the SVM, with the goal of better balancing the positive and negative classifications. From an analysis of several SVM thresholds on tuning data, we determined appropriate thresholds of 0.5 for terrorist events and 0.3 for disease outbreaks. These thresholded SVMs were iteratively self-trained and evaluated against the tuning data in each domain to determine the stopping criteria for the iterations. A new SVM was generated after each iteration, and the one selected was based on its F-score on event sentence labels. The reason for this is that recall on event sentences is an important characteristic desired from the classifier.

Table 6.4: Seed patterns used for self-training

Terrorist Events Seeds	Disease Outbreaks Seeds
<subject> exploded	<subject> imports
assassination of <np>	outbreak on <np>
death of <np>	$<\#^{th}$ case>
<subject> was kidnapped	<subject> was destroyed
murder of <np>	outbreak with <np>
caused <dirobj>	outbreak from <np>
bogota in <np>	<subject> killed people
<subject> was injured	crows in <np>
destroyed <dirobj>	<# source>
<subject> was located	<# crows >
responsibility for <np>	$<\#^{th}$ cases>
claimed <dirobj>	bse in <np>
<subject> was murdered	<# mosquito >
<subject> destroyed	crow in <np>
<subject> located	<# crow>
<subject> caused	<subject> spreading
attack against <np>	<subject> is outbreak
was taken to <np>	dengue in <np>
<subject> took place	died of <np>
attack in <np>	mosquito in <np>

Table 6.5 summarizes the results of this evaluation on tuning data. Recall that *iteration #0* in this table refers to the seeds that start off the self-training process. Again, we based our choice of classifier on its F-score on event sentences so as to achieve higher recall on event sentences. Accordingly, we find that the event sentence F-score of the classifier levels off (and starts degrading) after the fifth iteration for terrorist events, and after the second iteration for disease outbreaks. Thus, the classifiers generated at the end of these iterations, respectively, were selected as the sentential event recognizer in each domain. To determine the true performance of the selected classifiers, they were then evaluated on the human-annotated gold standard data sets. Table 6.6 presents the evaluation of these selected SVMs on the human-annotated data. Even though we selected our classifier based on its F-score on event sentences, we note that its overall accuracy and its F-score on nonevent sentences are also relatively high.

6.2.1.3 MIL. Another weakly supervised approach for training a classifier for sentential event recognition is to use a multiple instance learning strategy. This strategy can train a classifier for sentence classification, using only document-level labels. Here, we use the sMIL multiple instance learning framework of Bunescu and Mooney [11] to train an SVM classifier for using only a set of relevant and irrelevant documents. The sMIL framework is specifically designed for training from a sparse distribution of the positive examples, which is precisely the nature of our data set. We use the same documents that were employed

Table 6.5: Self-training iterations evaluated on tuning data

		Event			Nonevent		
	Acc	Pr	Rec	F	Pr	Rec	F
Terrorist Events (MUC-4)							
(Seeds) Iter #0	0.82	0.64	0.21	0.31	0.84	0.97	0.90
Iter #1	0.83	0.68	0.22	0.33	0.84	0.98	0.90
Iter #2	0.83	0.70	0.23	0.34	0.84	0.98	0.90
Iter #3	0.79	0.46	0.60	0.52	0.90	0.84	0.87
Iter #4	0.78	0.45	0.64	0.53	0.90	0.82	0.86
Iter #5	0.81	0.50	0.58	0.54	0.90	0.86	0.88
Iter #6	0.81	0.51	0.54	0.53	0.89	0.88	0.88
Iter #7	0.81	0.52	0.54	0.53	0.89	0.88	0.88
Disease Outbreaks (ProMed)							
(Seeds) Iter #0	0.67	0.80	0.11	0.20	0.66	0.98	0.79
Iter #1	0.69	0.59	0.50	0.55	0.74	0.80	0.77
Iter #2	0.65	0.52	0.62	0.56	0.75	0.67	0.71
Iter #3	0.44	0.37	0.78	0.50	0.66	0.25	0.36
Iter #4	0.46	0.38	0.77	0.51	0.68	0.29	0.41

Table 6.6: Evaluation of best Self classifiers on human annotations

			Event			Nonevent		
		Acc	Pr	Rec	F	Pr	Rec	F
Terrorist Events	SVM (0.5, iter #5)	0.83	0.48	0.73	0.58	0.94	0.85	0.89
Disease Outbreaks	SVM (0.3, iter #2)	0.65	0.37	0.70	0.49	0.88	0.64	0.74

previously in the self-training process (**Self**) for training — a total of 700 relevant and 600 irrelevant documents for terrorist events, and 2,000 relevant and 4,000 irrelevant documents for disease outbreaks.

Again, the SVM obtained at the end of this process needs to be appropriately thresholded to achieve a better balance of positive and negative examples identified by the classifier. Using the tuning data, -0.92 and -0.91 are determined as the best thresholds for the terrorist events and disease outbreaks, respectively. These thresholds are again determined based on their F-score on event sentences. The SVM generated from the sMIL technique is applied to the human-annotated test sentences (100 terrorist event documents and 120 disease outbreaks documents) to evaluate its sentence classification accuracy. These results are presented in Table 6.7.

6.2.1.4 Summary of the models. To enable a comparison of the three sentential

Table 6.7: Evaluation of best MIL classifiers on human annotations

			Event			Nonevent		
		Acc	Pr	Rec	F	Pr	Rec	F
Terrorist Events (MUC-4)	SVM (-0.92)	0.86	0.58	0.62	0.60	0.92	0.91	0.92
Disease Outbreaks (ProMed)	SVM (-0.91)	0.64	0.35	0.64	0.45	0.86	0.64	0.74

event recognizers (Anskey, Self and MIL) in the PIPER model, their evaluation on human annotations is summarized in Table 6.8. We observe that, despite using different amounts of supervision, the performance of the weakly supervised Self and MIL classifiers is not drastically different from the Anskey classifiers. Between the two data sets, the overall performance of the classifiers is higher on the terrorist events data, indicating a higher level of difficulty of event recognition in the disease outbreaks data.

On the terrorist event data set, the overall accuracy of all four classifiers is between 83% and 86%. Of the two Anskey classifiers, the SVM-based classifier is clearly superior in the terrorism domain, achieving a higher overall accuracy and higher F-scores on the two labels. Of the weakly supervised classifiers, the Self model achieves a recall of 0.73 on the terrorism event sentences, just 2% lower than the $\text{Anskey}_{\text{SVM}}$ model, but has a precision 6% lower. The MIL model, on the other hand, has a much higher precision of 0.58 on the terrorism event sentences, but at the cost of much lower recall of 0.62. Between the two weakly supervised classifiers there is an even precision-recall trade-off on the terrorism event sentences, with the self-trained classifier achieving greater recall. Overall, for the terrorist events data, $\text{Anskey}_{\text{SVM}}$ is the best classifier among the four, and illustrates the benefits of the greater supervision.

On the disease outbreaks data set the overall performance of all classifiers is somewhat lower, in the range 64% through 74%. Additionally, the trends in performance are different from those seen in the terrorist events data set. In this domain, between the two Anskey classifiers, the Naïve Bayes Anskey classifier achieves better overall performance (accuracy and F-scores on the two labels) than the SVM. This is likely due to the much smaller size of the disease outbreaks training data set, which may have a greater effect on the SVM. Even though $\text{Anskey}_{\text{NB}}$ achieves the best overall accuracy of 74%, the weakly supervised classifiers have a greater recall on the disease outbreaks event sentences. On this data

Table 6.8: Evaluation of event recognizers on human annotations

		Event			**Nonevent**		
	Acc	Pr	Rec	F	Pr	Rec	F
Terrorist Events (MUC-4)							
$\text{Anskey}_{\text{SVM}}$ (-0.82)	0.86	0.54	0.75	0.63	0.95	0.88	0.91
$\text{Anskey}_{\text{NB}}$ (0.5)	0.83	0.50	0.70	0.58	0.94	0.86	0.90
Self_{SVM} (0.5, iter #5)	0.83	0.48	0.73	0.58	0.94	0.85	0.89
MIL_{SVM} (-0.92)	0.86	0.58	0.62	0.60	0.92	0.91	0.92
Disease Outbreaks (ProMed)							
$\text{Anskey}_{\text{SVM}}$ (-0.57)	0.69	0.38	0.52	0.44	0.84	0.75	0.79
$\text{Anskey}_{\text{NB}}$ (0.5)	0.74	0.45	0.53	0.48	0.85	0.80	0.82
Self_{SVM} (0.3, iter #2)	0.65	0.37	0.70	0.49	0.88	0.64	0.74
MIL_{SVM} (-0.91)	0.64	0.35	0.64	0.45	0.86	0.64	0.74

set, the Self classifier has a 13% higher recall and 8% lower precision on event sentences compared to the $\text{Anskey}_{\text{NB}}$ classifier, resulting in a slightly (1%) higher F-score. Going by this measure, the self-trained classifier is the best in this domain. On the other hand, going by overall accuracy (and precision on event sentences), the $\text{Anskey}_{\text{NB}}$ classifier is found to do well for disease outbreaks.

Table 6.9 presents a slightly different perspective for evaluating classifier performance. Each of these classifiers identifies event sentences for the localized text extraction model, and discards nonevent sentences. The extraction model can then focus on extracting event role fillers only from the event sentences identified by the classifiers. Thus, from the point of view of the extraction model, a higher proportion of event sentences within the classified data set can help improve extraction precision, since the extraction model would be less distracted by the irrelevant information in nonevent sentences. On the other hand, any event sentences incorrectly classified as nonevent sentences negatively affects the extraction model, since it is unable to extract role fillers from the discarded sentences. These two properties of the classified data are captured in Table 6.9. The *Nonevent Recall* and *Event Recall* columns in the **Classifier** side of the table list the recall of the classifiers on the two labels (taken directly from Table 6.8). The *EvSent Ratio* column in the **Data** side of the table lists the proportion of event sentences that remain after filtering the sentences labeled as nonevent sentences by the classifier. The *EvSent Lost* column lists the percentage of event sentences erroneously classified as nonevent sentences (i.e., discarded) by the classifier.

The "All EvSent" rows in both domains indicate the original state of the data, when no classifier is applied. This is equivalent to applying a classifier that classifies all sentences as event sentences. We observe from the original distribution of the data that only 16% of the

Table 6.9: Classifier effect on data sets

Classifier			**Data**	
	Event Recall	*Nonevent Recall*	*EvSent Ratio*	*EvSent Lost*
Terrorist Events (MUC-4)				
All EvSent	1.00	0.00	16%	0%
$\text{Anskey}_{\text{SVM}}$ (-0.82)	0.75	0.88	55%	25%
$\text{Anskey}_{\text{NB}}$ (0.5)	0.70	0.86	48%	30%
Self_{SVM} (0.5, iter #5)	0.73	0.75	36%	27%
MIL_{SVM} (-0.92)	0.62	0.91	56%	38%
Disease Outbreaks (ProMed)				
All EvSent	1.00	0.00	23%	0%
$\text{Anskey}_{\text{SVM}}$ (-0.57)	0.52	0.75	39%	48%
$\text{Anskey}_{\text{NB}}$ (0.5)	0.53	0.80	44%	47%
Self_{SVM} (0.3, iter #2)	0.70	0.64	36%	30%
MIL_{SVM} (-0.91)	0.64	0.64	35%	36%

sentences are event sentences in the terrorist event data, and only 23% of the sentences are event sentences in the disease outbreaks data. The proportion of event sentences changes significantly after applying our sentential event classifiers. In the terrorist events data, the MIL classifier changes proportion of event sentences to almost 56%. But this is at the cost of 38% of event sentences incorrectly discarded (classified as nonevent). The $\text{Anskey}_{\text{SVM}}$, on the other hand, achieves a similar event sentence proportion of 55%, at the cost of losing only 25% of the event sentences. On the disease outbreaks data, we find that both Anskey classifiers are quite aggressive, with $\text{Anskey}_{\text{NB}}$ losing almost 47% of the event sentences, in improving the event sentence proportion from 23% to about 44%. In this analysis, the weakly supervised Self classifier appears to be much better, losing only 30% of the event sentences and improving the event sentence proportion from 23% to 36%.

From this analysis, the $\text{Anskey}_{\text{SVM}}$ classifier appears to be the best sentential event recognizer for the terrorist events IE task, and the Self_{SVM} classifier appears to be the best for the disease outbreaks IE task.

6.2.2 The Glacier Model

Chapter 4 (Section 4.2) describes sentential models for estimating the probabilities of event sentences in given documents. A probabilistic Naïve Bayes approach and a normalized SVM approach are presented for obtaining these probabilities. Both approaches estimate the probability of each sentence being an event or nonevent sentence based on features associated with the sentence. These models are exactly the same as the Anskey classifiers in the Piper model. However, unlike the sentential event recognizers of the Piper model, these classifiers are not used for classification, but for generating probability estimates of event sentences. An evaluation of these probability estimates would require human judgements of probability estimates on sentences in the data set. Such human estimates are much harder to obtain, as compared to event/nonevent labels, which makes it difficult to evaluate the estimates from the classifiers. A rough idea of their performance was seen in the previous section evaluating the sentential event recognizers of the Piper model. In that evaluation, the thresholded Anskey classifiers were evaluated on human annotations. It was found that the SVM performed well on the terrorist events data set, while the Naïve Bayes classifier worked well on the disease outbreaks data. The next section evaluates the full-fledged IE systems and provides an indirect assessment of probability values from the Glacier sentential event recognizers.

6.3 IE Evaluation Data and Methodology

Having evaluated the sentential components of our two models for IE, we can now employ these within the complete IE system for event role identification. The models are applied to test data to empirically illustrate the need in IE for event description information in the wider context. The evaluation serves to show the benefit of using a unified probabilistic model over a pipelined approach for incorporating information from the wider context. Our models for IE are also compared against other IE systems that rely only on local contextual information for identifying event role fillers in text.

Like the sentential event recognizers, the IE models are evaluated on two types of events — *terrorist events* and *disease outbreaks*. The test data consist of 200 documents from the MUC-4 data set for terrorist events and 120 ProMed documents for disease outbreaks. All of these documents come with corresponding IE answer key templates. The task for terrorist event (MUC-4) IE is to identify five types of event role fillers from reports about terrorist events: *weapon*, *perpetrator organization*, *perpetrator individual*, *victim* and *physical target*. For the disease outbreaks domain, the IE models are required to extract two types of event role fillers from the test documents: *disease*, *victim*. Recall that an answer key template contains event role filler strings from the corresponding document. A document may have one or more event templates for the various event descriptions within it. A document may have no event templates if it does not contain any relevant events. The IE systems are evaluated on all of the test documents.

The complete IE task (the *Scenario Template* task) requires the IE system to create event templates, one template per incident. Template generation is a complex process, requiring coreference resolution and discourse analysis to determine how many incidents were reported and which facts belong with each incident. Our work focuses on extraction of event role strings and not template generation, so we evaluated our IE models directly on the extractions themselves, before template generation would take place. For example, suppose that the word *grenade* is extracted as the weapon used in a terrorist event. This extraction is presumed to be a correct extraction if any of the answer key templates associated with this document contains *grenade* as the role filler for the weapon event role. This approach directly measures how accurately the models identify event role information, without confounding factors from the template generation process. For example, if a coreference resolver incorrectly decides that two extractions are coreferent and merges them, then only one extraction would be scored. We use a head noun scoring scheme, where an extraction is considered to be correct if its head noun matches the head noun in the answer

key.[3] Also, pronouns were discarded from both the system responses and the answer keys since no coreference resolution is done. Duplicate extractions (e.g., the same string extracted by different patterns) are conflated before being scored, so they count as just one hit or one miss.

A scoring program implementing these extraction scoring rules is used to compute the precision, recall and F-score for the various IE models. The precision of an IE system can be computed based on the fraction of extracted role filler strings correctly matched to the answer key templates. Similarly, the coverage or recall of the system can be calculated based on the fraction of strings in the answer key templates that were extracted by the system. The precision, the recall and their harmonic mean (the F-score) over a test data set are used as the measure of success of the IE models.

6.4 Baselines for IE Evaluation

To determine whether our IE models perform better than existing approaches, we compare their performance on two IE tasks to that of baseline IE systems that rely only on the local context surrounding the candidate extractions. The baselines include an existing pattern-based IE system and a classifier-based system. Additionally, an extraction system that simply uses the semantic classes of words is evaluated as a third baseline.

6.4.1 AutoSlog-TS

The first baseline used in this evaluation is an existing pattern-based IE system called AutoSlog-TS [92]. This system uses lexico-syntactic extraction patterns for identifying event role fillers in text. The patterns are learned in a weakly supervised manner from a set of relevant and irrelevant documents. Figure 6.1 presents an overview of the AutoSlog-TS system. The system first generates an exhaustive set of extraction patterns, such that they literally extract every noun phrase in the relevant and irrelevant documents. The types of patterns generated by AutoSlog-TS are listed in Appendix C. The primary component of this system is the pattern ranking module which ranks the exhaustive set of extraction patterns based on their likelihood of occurring within a relevant document. It uses a ranking metric called *rlogf* for this purpose. A human expert then examines the patterns near the top of the list and selects the best patterns for the task at hand. Additionally, the human expert is also tasked with separating the patterns into groups based on the event role fillers they extract. Thus at the end of this process, the system obtains extraction patterns for

[3] For example, "armed men" will match "five armed men."

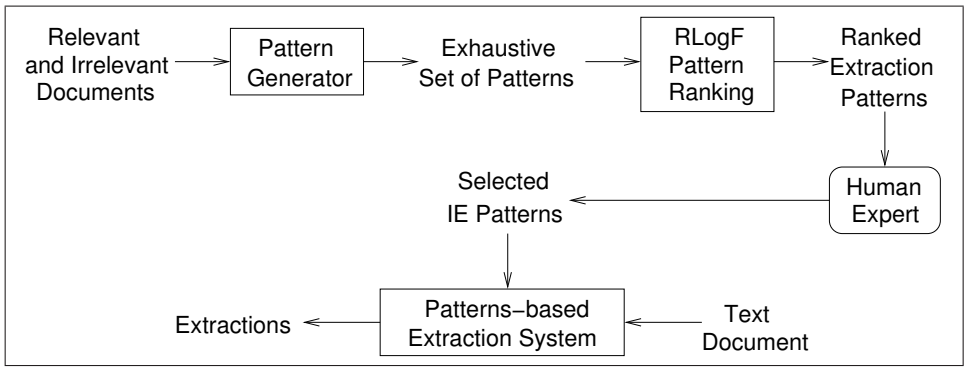


Figure 6.1: Overview of the AutoSlog-TS IE system

each event role in the IE task. These IE patterns are then applied by the system to unseen text documents to extract event role fillers from them.

The AutoSlog-TS system is freely available software, which comes trained for the MUC-4 terrorist event IE task, and for the ProMed IE task of extracting information about disease outbreaks. The system consists of 600 extraction patterns for the MUC-4 terrorist events, and 111 extraction patterns for the ProMed disease outbreaks IE task. Table 6.10 presents the IE evaluation of this system on test sets in the two domains. The table includes the aggregate performance at this task, and the performance of the system broken down by event role. Because we do not do template generation, these scores cannot be directly compared to existing IE systems that have been designed for these data sets. However, with this caveat, these performance numbers are in the same ballpark as those achieved by other IE systems.[4] So, this baseline is indeed a competent one.

6.4.2 Unconstrained Naïve Bayes Extractor

The unified probabilistic model for IE presented in this research consists of a phrasal component called the *Plausible Role-Filler Recognizer*, which is conditioned on event sen-

Table 6.10: AutoSlog-TS evaluation on the test data

Terrorist Events (MUC-4)				***Disease Outbreaks (ProMed)***			
Event Role	*Pr*	*Rec*	*F*	*Event Role*	*Pr*	*Rec*	*F*
PerpInd	0.33	0.49	0.40	Disease	0.33	0.60	0.43
PerpOrg	0.52	0.33	0.41	Victim	0.36	0.49	0.41
Target	0.54	0.59	0.56	**Aggregate**	0.35	0.53	0.42
Victim	0.48	0.54	0.51				
Weapon	0.38	0.45	0.41				
Aggregate	0.45	0.50	0.47				

[4]For a summary of MUC-4 performance scores, see the work by Chieu et al. [15].

tences so as to incorporate event information from the wider sentential context. If the effect of the sentential component were to be eliminated from this model, the plausible role-filler recognizer would then become an unconditioned classifier-based extraction system, which relies on the local contextual evidence surrounding the phrases for recognizing event role fillers. Thus, as our second baseline, we train a Naïve Bayes IE classifier (unconditioned NB) that is analogous to the plausible role-filler recognizer in the GLACIER IE model, except that this baseline system is not conditioned on the assumption of having event sentence information. Consequently, this unconditioned NB classifier is akin to a traditional supervised learning-based IE system that uses only local contextual features to make extraction decisions. Formally, the unconditioned NB classifier uses the formula:

$$\begin{aligned} & P(PlausFillr(NP_i)|F) \\ & = \frac{1}{P(F)} P(PlausFillr(NP_i)) * \prod_{f_i \in F} P(f_i|PlausFillr(NP_i)) \end{aligned} \tag{6.1}$$

where F is the set of local features, $P(F)$ is the normalizing constant, the product term in the equation is the likelihood, and $P(PlausFillr(NP_i))$ is the prior probability, which is obtained from the ratio of the class labels in the training data. For this unconditioned NB system we again use the Weka Toolkit [122] implementation of Naïve Bayes, with the default classifier configuration. The features used in this system are obtained in exactly the same way (described in Chapter 4, Section 4.4) as they were for GLACIER's plausible role-filler recognizer.

The unconditioned NB classifier is trained using the answer key templates associated with the training documents for the MUC-4 and ProMed data sets. The answer keys are used to create approximate annotations of noun phrases as being a filler for an event role or not. We identify all instances of each answer key string in the source document and consider these as positive training examples. All other noun phrases in the document that do not match an answer string in the answer key template are considered as negative examples for training. This produces noisy training data, however, because some instances occur in undesirable contexts. For example, if the string "man" appears in an answer key as a victim, one instance of "man" in the text document may refer to the actual victim in an event sentence, while another instance of "man" may occur in a nonevent context (e.g., background information) or may refer to a completely different person. Using these approximate annotations, unconditioned NB classifiers are trained for extracting event role fillers for the MUC-4 terrorist events, and for the ProMed disease outbreaks IE tasks.

The trained unconditioned NB classifiers are applied to the test data in the two domains, and evaluated against the answer key templates on their precision, recall and F-score. For the Naïve Bayes classifier, the natural threshold for distinguishing between positive and negative classes is 0.5, but we also evaluated the classifier with thresholds of 0.7 and 0.9 to see if we could effect a recall/precision tradeoff. Table 6.11 presents the results of the unconditioned NB systems. The classifier performs comparably to the AutoSlog-TS baseline on most event roles, although a threshold of 0.90 is needed to reach comparable performance on the disease outbreaks data. The relatively low numbers across the board indicate that these corpora are challenging, but these results suggest that our plausible role-filler recognizer is competitive with other IE systems.

6.4.3 Semantic Class Extractor

Both of the models for IE presented in the dissertation (PIPER and GLACIER) advocate the need for evidence from the wider context to enable the use of local contextual information for identifying event role fillers. By recognizing event descriptions in the wider context, much weaker local contextual clues can now be useful for identifying event role fillers in text. If the semantics of an entity match those expected of an event role filler, then within the context of an event description this evidence can sometimes be sufficient to extract that entity as an event role filler. For example, if we see a weapon word (like *gun*) mentioned within the context of a terrorist event description, this information can be sufficient to extract the weapon word as the instrument used in the terrorist event. Thus, we implement a baseline extraction system that simply extracts certain semantic class words as event role fillers, if they appear in an event sentence context.

This baseline system uses the two-stage pipelined approach of the PIPER model. It

Table 6.11: Unconditioned NB evaluation on the test data

	NB 0.5			NB 0.7			NB 0.9		
Event Role	*Pr*	*Rec*	*F*	*Pr*	*Rec*	*F*	*Pr*	*Rec*	*F*
Terrorist Events (MUC-4)									
PerpInd	0.36	0.34	0.35	0.41	0.25	0.31	0.51	0.17	0.25
PerpOrg	0.35	0.46	0.40	0.43	0.31	0.36	0.56	0.15	0.24
Target	0.53	0.49	0.51	0.58	0.42	0.48	0.67	0.30	0.41
Victim	0.50	0.50	0.50	0.58	0.37	0.45	0.75	0.23	0.36
Weapon	1.00	0.05	0.10	1.00	0.04	0.07	1.00	0.02	0.04
Aggregate	0.55	0.37	0.44	0.60	0.28	0.38	0.70	0.17	0.27
Disease Outbreaks (ProMed)									
Disease	0.20	0.73	0.31	0.23	0.67	0.34	0.34	0.59	0.43
Victim	0.29	0.56	0.39	0.37	0.52	0.44	0.47	0.39	0.43
Aggregate	0.25	0.65	0.36	0.30	0.60	0.40	0.41	0.49	0.44

first applies a sentential event recognizer component to the input text, and then extracts event role fillers from the event sentences based on their semantic classes. To decide which semantic classes to extract as event role fillers, this baseline uses the semantic class to event role mapping that were created for the semantic affinity model (described in Chapter 3, Section 3.4.2). Recall that this semantic class mapping is based on a human expert's knowledge of the event domains. Based on real world knowledge of event contexts, semantic classes that tend to be role fillers for specific event roles are mapped to those event roles. For example, buildings are usually targets of terrorist attacks, so the *building* semantic class is mapped to the *physical target* event role in terrorist events. Using this mapping, event role fillers are extracted by this baseline from event sentences.

The semantic classes of words and phrases in text are identified using the semantic class tagger included in the Sundance NLP system [95]. This tagger is based on dictionary lookups in a semantic dictionary. The Sundance system consists of semiautomatically created dictionaries of about 65,200 words, of which about 7,600 words appear in a domain-independent dictionary, 1,200 words appear in a terrorism-specific dictionary, and the remaining 56,400 words[5] appear in a biomedical dictionary. The nouns in this dictionary have semantic classes associated with them from a set of about 171 semantic classes. The event sentences in this experiment are obtained using the **Anskey** sentential event recognizer, which is a supervised classifier trained on approximate sentence annotations using the IE answer key templates. Based on the performance scores of the variations of this classifier presented in Section 6.2, we use the SVM (-0.82) Anskey classifier for the terrorist events data, and the NB (0.5) Anskey for the disease outbreaks data. Using the sentential event recognizer, the Sundance-based semantic tagger, and the semantic class to event role mapping, this baseline system extracts event role fillers from event sentences in a given document.

Table 6.12 shows the results of this system on the two test data sets. Observe that on

Table 6.12: Semantic class baseline evaluation on the test data

Terrorist Events (MUC-4)				***Disease Outbreaks (ProMed)***			
Event Role	*Pr*	*Rec*	*F*	*Event Role*	*Pr*	*Rec*	*F*
PerpInd	0.34	0.46	0.39	Disease	0.12	0.70	0.20
PerpOrg	0.30	0.55	0.39	Victim	0.12	0.65	0.20
Target	0.24	0.78	0.36	**Aggregate**	0.12	0.68	0.20
Victim	0.18	0.48	0.26				
Weapon	0.34	0.69	0.45				
Aggregate	0.24	0.57	0.34				

[5] A large number of biomedical terms in this dictionary, like diseases and symptoms, have been obtained automatically from the UMLS [66] ontology.

three of the terrorism event roles — *PerpInd*, *PerpOrg* and *Weapon* — this simple strategy matches the performance of the pattern-based AutoSlog-TS system. In fact, the *Weapon* event role clearly surpasses the AutoSlog-TS baseline. However, on the remaining event roles there is a sharp performance drop in comparison with AutoSlog-TS. In these cases, the semantic extractions get a high recall, but are extremely poor on precision. Note that the event roles that match AutoSlog-TS in performance are those whose role fillers are closer together in the semantic space, and more clearly distinguishable from other semantic classes. For example, perpetrator words like *assailant*, *terrorist* and *militant* are strongly associated with perpetrators of terrorism, and are clearly distinguishable from the other role fillers. Similarly, weapon words like *grenade*, *explosive* and *rifle* are strongly associated with the weapon event role and can be easily told apart from the other event roles. The victim event role fillers on the other hand can be filled by any of the human entities mentioned in the text, such as *the reporter*, *chairman* and *president*, who could just as easily play any other event roles (or no event roles) and are not particularly tied to the victim event role. The semantic notion of this event role is, therefore, somewhat nebulous and requires more contextual evidence to enable a clearer delineation of "victims" in the semantic space. Similarly, the disease event role fillers are names of diseases, but not all mentions of disease names appear as an outbreak. Some diseases are ongoing infections, others are mentioned to compare symptoms with an outbreak. A common occurrence we found in these data was the extraction of general terms such as "an illness," "a virus" and "a disease" as disease terms. None of these cases satisfy the definition of an outbreak, and require additional contextual information to filter out the incorrect role fillers. This illustrates that only relying on semantics of the potential role fillers is not sufficient for their extraction from event sentences.

6.5 PIPER Model Evaluation

Chapter 3 describes several configurations of the PIPER model for event role extraction that combine variations of the sentential event recognizer with variations of a pattern-based localized text extraction module, organized in a pipeline. All of these configurations consist of a sentential event recognizer that identifies event sentences in text, followed by a pattern-based text extraction component applied to the event sentences identified by the sentential component. We evaluate these IE models here on the two domains, and compare these results against those of the baseline IE systems.

6.5.1 PIPER$_{\text{Anskey/LexAff}}$

This configuration of the PIPER model consists of the *Anskey* approach for the sentential event recognizer and the *LexAff* approach for the localized text extraction module. The sentential event recognizer uses a supervised classifier, trained on approximate annotations from the answer key templates, for identifying event sentences in text. The localized text extraction component is a pattern-based extraction system that learns extraction patterns using a lexical affinity metric computed from answer strings in the answer key templates. This combination in the pipeline requires the most supervision of the various PIPER configurations, since both components use IE answer key templates in their training strategies.

The sentential event recognizer used for event sentence identification in the terrorism domain is an SVM thresholded at -0.82, and in the disease outbreaks domain is a Naïve Bayes classifier thresholded at 0.5. These classifiers and their corresponding thresholds were selected based on the evaluation of the sentential event recognizers described earlier in Section 6.2. The classifiers were applied to text documents to identify a subset of sentences that contain event descriptions. The localized text extraction component then provided ranked patterns for identifying event roles. The performance of this IE model was measured based on its accuracy of the event role fillers extracted.

The patterns learned by the localized text extraction component were obtained by first generating an exhaustive set of all possible extraction patterns in the training data using the AutoSlog pattern generation system. All patterns that appeared three times or less in the training data were discarded. Additionally, a lower probability threshold θ_l of 0.5 was applied to discard those patterns whose probability of appearing in relevant documents was 0.5 or lower. The remaining patterns were then ranked using the *lexical affinity* metric, which uses answer key strings to identify appropriate patterns for use within the event sentences. A separate ranking of these patterns was created for each event role to be extracted. Top N patterns, for chosen values of N, were then selected for each event role and applied to the event sentences to extract event role fillers.

The parameter N, therefore, needed to be determined for each event role, before the model could be evaluated on the test data. This parameter was determined in our experiments by applying the model to tuning data in each domain, and measuring IE performance for a range of values of N. Table 6.13 shows these results on the tuning set (200 documents for terrorist events, 20 documents for disease outbreaks) for values of N ranging from 50 through 250. Each set of top N patterns was applied only to the event sentences identified

Table 6.13: Parameter tuning for PIPER$_{\text{Anskey/LexAff}}$

Terrorist Events (MUC-4)															
	PerpInd			**PerpOrg**			**Target**			**Victim**			**Weapon**		
N	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
50	.32	.13	.18	.38	.11	.18	.50	.43	.46	.49	.40	.44	.73	.45	.56
100	.29	.23	.25	.44	.20	.27	.41	.46	.43	.45	.44	.44	.69	.49	.57
150	.27	.23	.25	.41	.21	.28	.41	.52	.46	.40	.46	.43	.64	.51	.57
200	.26	.26	.26	.30	.32	.31	.39	.56	.46	.35	.47	.40	.55	.53	.54
250	.25	.30	.27	.30	.32	.31	.39	.56	.46	.36	.49	.41	.55	.53	.54

Disease Outbreaks (ProMed)						
	Disease			**Victim**		
N	*P*	*R*	*F*	*P*	*R*	*F*
50	.50	.32	.39	.67	.31	.43
100	.35	.37	.36	.60	.38	.46
150	.33	.37	.35	.50	.41	.45
200	.33	.37	.35	.50	.41	.45
250	.30	.37	.33	.50	.44	.47

by the sentential event recognizer. Extractions from these sentences were evaluated against the role fillers listed in the answer key templates. The precision, recall and F-score was computed for each value of N across the various event roles. Based on this experiment, the values of N selected for each event role are shown by the highlighted cells in Table 6.13.

To enable a comparison of our IE model with other systems and with the baseline systems, our selected model was then evaluated on the test data, which consist of 200 MUC-4 documents for terrorist events and 120 ProMed documents for disease outbreaks, along with their answer key templates. The evaluation of the PIPER$_{\text{Anskey/LexAff}}$ model on the official test data is presented in the highlighted rows of Table 6.14. The rows labeled "PIPER$_{\text{AR}}$-All" present the results of applying the lexical affinity patterns to the entire test set, without the use of the sentential event recognizer. These results show the

Table 6.14: PIPER$_{\text{Anskey/LexAff}}$ evaluation on the test data

Terrorist Events (MUC-4)															
	PerpInd			**PerpOrg**			**Target**			**Victim**			**Weapon**		
Model	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
PIPER$_{\text{AR}}$-All	.19	.43	.27	.36	.51	.42	.43	.45	.44	.38	.55	.44	.44	.55	.49
PIPER$_{\text{AR}}$-Rel	.26	.38	.31	.46	.45	.46	.54	.45	.49	.49	.49	.49	.48	.53	.51
PIPER$_{\text{AR}}$-Sel	.26	.38	.31	.46	.45	.45	.54	.45	.49	.48	.50	.49	.48	.55	.51
AutoSlog-TS	.33	.49	.40	.52	.33	.41	.54	.59	.56	.48	.54	.51	.38	.45	.41
NB-Ex	.36	.34	.35	.35	.46	.40	.53	.49	.51	.50	.50	.50	1.00	.05	.10
Sem-Ex	.34	.46	.39	.30	.55	.39	.24	.78	.36	.18	.48	.26	.34	.69	.45

Disease Outbreaks (ProMed)						
	Disease			**Victim**		
Model	*P*	*R*	*F*	*P*	*R*	*F*
PIPER$_{\text{AR}}$-All	.38	.54	.45	.28	.48	.35
PIPER$_{\text{AR}}$-Rel	.43	.51	.47	.33	.43	.38
PIPER$_{\text{AR}}$-Sel	.40	.52	.45	.30	.45	.36
AutoSlog-TS	.33	.60	.43	.36	.49	.41
NB-Ex	.34	.59	.43	.47	.39	.43
Sem-Ex	.12	.70	.20	.12	.65	.20

performance of the lexical affinity patterns by themselves, without the support provided by event sentences. On the other hand, the rows labeled "PIPER$_{AR}$-Rel" enumerate the results of the case where the patterns are applied only to event sentences (from the sentential event recognizer). These results are the true results of our pipelined approach. Finally, the "PIPER$_{AR}$-Sel" rows present results of our enhanced version of the pipelined model, which separates *primary* patterns from *secondary patterns*. In this enhancement, the top N patterns are separated into two sets — primary and secondary patterns — based on threshold θ_u applied to their probability of appearing in relevant documents. Patterns with a probability greater than 0.8 are considered primary patterns, and those with a probability of 0.8 or less are considered secondary patterns. The primary patterns are applied to the entire tuning data set, while the secondary patterns are applied only to the event sentences identified by the sentential component. This enhancement to the pipelined model allows certain patterns representing extremely strong local evidence to override the event sentence information.

Looking at the PIPER$_{AR}$-All rows, we see relatively good recall on most event roles, but the precision is usually low. Comparing PIPER$_{AR}$-All with PIPER$_{AR}$-Rel, we observe that in every case precision improves, demonstrating that our sentence classifier is having the desired effect. However, the precision gain comes with some loss in recall points. By selectively applying patterns in the PIPER$_{AR}$-Sel method, we see some of the lost recall gained back in four of the seven event roles. However, a comparison of the F-scores shows that the PIPER$_{AR}$-Rel model is preferred over the other two PIPER$_{Anskey/LexAff}$ models.

For comparison, results from the baseline IE systems are presented in the rows below the highlighted rows. The rows labeled "AutoSlog-TS" contains the results of the AutoSlog-TS IE system. The rows labeled "NB-Ex" contain the results of the unconstrained Naïve Bayes classifier. The rows labeled "Sem-Ex" contain the results of the semantic class extractor. Comparing the PIPER$_{Anskey/LexAff}$ model with the baselines, we see that **PerpOrg**, **Weapon** and **Disease** event roles get a better F-score than all of the baselines. The **PerpInd** event role in the terrorist events data and the **Victim** event role in the disease outbreaks data have F-scores lower than the baselines, primarily due to lower precision. On the other hand, the **Target** and **Victim** event roles in the terrorist events data have F-scores slightly lower than the baseline primarily due to their lower recall on the IE task.

6.5.2 PIPER$_{Self/SemAff}$

This configuration of the PIPER model consists of the *Self* approach for the sentential event recognizer and the *SemAff* approach for the localized text extraction module. The

sentential event recognizer uses a self-trained classifier, trained on a set of relevant documents, irrelevant documents and a set of seeds. The localized text extraction component is a pattern-based extraction system that learns extraction patterns using a *semantic affinity* metric computed from relevant and irrelevant documents. This combination in the pipeline requires less supervision, since both components require only document-level relevance annotations, and some seed extraction patterns for the classifier.

The sentential event recognizer used for event sentence identification in the terrorism domain was an SVM thresholded at 0.5, and in the disease outbreaks domain was an SVM thresholded at 0.3. These classifiers and their corresponding thresholds were selected based on the evaluation of the sentential event recognizers described earlier in Section 6.2. The classifiers were tasked with identifying event sentences in the text documents provided to them. The localized text extraction component then used extraction patterns for identifying event roles within the classified event sentences. The performance of this IE model was measured based on its accuracy of the event role fillers extracted.

The patterns learned by the localized text extraction component were obtained by first generating an exhaustive set of all possible extraction patterns from the training data using the AutoSlog system. All patterns that appeared three times or less in the relevant and irrelevant documents were discarded. Additionally, a lower probability threshold θ_l of 0.5 was applied to discard those patterns whose probability of appearing in relevant documents was 0.5 or lower. The remaining patterns were then ranked using the *semantic affinity* metric, which uses relevant and irrelevant documents along with a manually created mapping between semantic classes and event roles. A separate ranking of these patterns was generated for each event role using the semantic affinity metric. The top-ranked patterns from these ranked lists were then applied to the event sentences identified by the sentential component. Like before, parameter N — the top-ranked patterns selected for each event role — was determined using the tuning data set in each domain. The top N ranked patterns for each event role were then selected based on our parameter tuning.

The selected model was then evaluated on the official test data for comparison with the baseline systems and other IE models. The evaluation of the $\text{PIPER}_{\text{Self/SemAff}}$ model on the official test data is presented in the highlighted rows of Table 6.15. The rows labeled "PIPER_{SS}-All" present the results of applying the semantic affinity patterns to all sentences in the test set. The rows labeled "PIPER_{SS}-Rel" enumerate the results of the case where the patterns are applied only to event sentences (from the self-trained sentential event recognizer). These results are the true results of our pipelined approach. Finally, the

Table 6.15: PIPER$_{\text{Self/SemAff}}$ evaluation on the test data

Terrorist Events (MUC-4)															
	PerpInd			**PerpOrg**			**Target**			**Victim**			**Weapon**		
Model	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
PIPER$_{\text{SS}}$-All	.28	.45	.34	.36	.51	.42	.37	.58	.45	.41	.42	.42	.36	.64	.46
PIPER$_{\text{SS}}$-Rel	.40	.42	.41	.47	.39	.43	.45	.52	.48	.52	.36	.42	.48	.55	.51
PIPER$_{\text{SS}}$-Sel	.40	.44	.42	.47	.40	.43	.45	.53	.48	.52	.39	.45	.45	.57	.50
PIPER$_{\text{AR}}$-Rel	.26	.38	.31	.46	.45	.46	.54	.45	.49	.49	.49	.49	.48	.53	.51
PIPER$_{\text{AR}}$-Sel	.26	.38	.31	.46	.45	.45	.54	.45	.49	.48	.50	.49	.48	.55	.51
AutoSlog-TS	.33	.49	.40	.52	.33	.41	.54	.59	.56	.48	.54	.51	.38	.45	.41
NB-Ex	.36	.34	.35	.35	.46	.40	.53	.49	.51	.50	.50	.50	1.00	.05	.10
Sem-Ex	.34	.46	.39	.30	.55	.39	.24	.78	.36	.18	.48	.26	.34	.69	.45

Disease Outbreaks (ProMed)						
	Disease			**Victim**		
Model	*P*	*R*	*F*	*P*	*R*	*F*
PIPER$_{\text{SS}}$-All	.32	.59	.41	.34	.53	.41
PIPER$_{\text{SS}}$-Rel	.33	.58	.42	.35	.53	.42
PIPER$_{\text{SS}}$-Sel	.33	.58	.42	.35	.53	.42
PIPER$_{\text{AR}}$-Rel	.43	.51	.47	.33	.43	.38
PIPER$_{\text{AR}}$-Sel	.40	.52	.45	.30	.45	.36
AutoSlog-TS	.33	.60	.43	.36	.49	.41
NB-Ex	.34	.59	.43	.47	.39	.43
Sem-Ex	.12	.70	.20	.12	.65	.20

"PIPER$_{\text{SS}}$-Sel" rows present results of our enhanced version of the pipelined model, which separates *primary* patterns from *secondary patterns*, then applies primary patterns to all sentences and secondary patterns only to the event sentences. Patterns with a probability greater than 0.8 are considered primary patterns, and those with a probability of 0.8 or less are considered secondary patterns. This enhancement to the pipelined model allows certain patterns representing extremely strong local evidence to override the event sentence information. For comparison, the evaluation results of the PIPER$_{\text{Anskey/LexAff}}$ model and the baseline models are presented in the rows below the highlighted rows.

We observe that the PIPER$_{\text{Self/SemAff}}$ model has greater IE coverage. Compared to the PIPER$_{\text{Anskey/LexAff}}$ model, it achieves a higher recall on almost all event roles except **PerpOrg** and **Victim**. Additionally, we see that two event roles — **PerpInd** in the terrorist events data set, and **Victim** in the disease outbreak data — show an improvement in performance as a result of the greater recall. Comparing PIPER$_{\text{SS}}$-All with PIPER$_{\text{SS}}$-Rel, we observe that precision improves in every case. However, in the disease outbreaks data set the precision improvement is quite small (1% on each event role), indicating that the self-trained SVM is eliminating relatively fewer nonevent sentences in this domain. By selectively applying patterns in the PIPER$_{\text{SS}}$-Sel method, we see small improvements in recall over PIPER$_{\text{SS}}$-Rel with no precision loss on almost all event roles in the terrorist events data set (except **Weapon**). The PIPER$_{\text{SS}}$-Sel model appears to have no change in performance over PIPER$_{\text{SS}}$-Rel on the disease outbreaks data. Overall, however, a comparison of the

F-scores shows that the PIPER$_{SS}$-Sel model is preferred over the other two PIPER$_{Self/SemAff}$ models. Compared with the baseline IE models, the PIPER$_{Self/SemAff}$ model closely matches the performance of the best baselines in the disease outbreaks IE task, while it surpasses the baselines for the **PerpInd**, **PerpOrg** and **Weapon** event roles (by 2%, 2% and 6%, respectively) for the terrorism IE task. This result is notable, because PIPER$_{Self/SemAff}$ is fully automated and uses less supervision as compared to the best baselines.

6.5.3 PIPER$_{MIL/SemAff}$

In addition to the self-trained SVM classifier for event sentence identification, a multiple instance learning approach was also explored for the sentential event recognizer. In this configuration of the PIPER model, the SVM for sentence classification, trained using a multiple instance learning strategy (MIL), is paired with the semantic affinity patterns (SemAff) for identifying event role fillers in event sentences. Like the self-trained SVM, the MIL approach also aims to use lower amounts of supervision in the form of relevant and irrelevant documents.

The sentential event recognizer used for event sentence identification in the terrorism domain was an SVM thresholded at -0.92, and the one in the disease outbreaks domain was an SVM thresholded at -0.91. The classifiers and their corresponding thresholds were selected based on the evaluation of the sentential event recognizers described earlier in Section 6.2. Similarly, the top N semantic affinity patterns were chosen based on the parameter tuning approach described in the previous sections. With the two components trained and tuned, the PIPER$_{MIL/SemAff}$ model for IE was then evaluated on the test sets.

The highlighted rows in Table 6.16 contain the evaluation of this IE model. As in the previous PIPER model evaluations, the rows labeled "PIPER$_{MS}$-All" present the results of applying the semantic affinity patterns to all sentences in the test set. The rows labeled "PIPER$_{MS}$-Rel" enumerate the results of the case where the patterns are applied only to event sentences (from the MIL sentential event recognizer). Finally, the "PIPER$_{MS}$-Sel" rows present results of our enhanced version of the pipelined model, which separates *primary* patterns from *secondary patterns*, then applies primary patterns to all sentences and secondary patterns only to the event sentences. Patterns with a probability greater than 0.8 are considered primary patterns, and those with a probability of 0.8 or less are considered secondary patterns. For comparison, the evaluation results of the PIPER$_{Anskey/LexAff}$ model, the PIPER$_{Self/SemAff}$ model and the baseline models are presented in the rows below the highlighted rows.

Since PIPER$_{MIL/SemAff}$ uses similar amounts of supervision as the PIPER$_{Self/SemAff}$, and

Table 6.16: PIPER$_{\text{MIL/SemAff}}$ evaluation on the test data

Terrorist Events (MUC-4)															
	PerpInd			**PerpOrg**			**Target**			**Victim**			**Weapon**		
Model	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
PIPER$_{\text{MS}}$-All	.28	.45	.34	.36	.51	.42	.37	.58	.45	.41	.42	.42	.36	.64	.46
PIPER$_{\text{MS}}$-Rel	.44	.38	.41	.45	.34	.39	.44	.46	.45	.55	.37	.44	.51	.53	.52
PIPER$_{\text{MS}}$-Sel	.44	.41	.42	.47	.40	.43	.44	.51	.47	.55	.39	.45	.48	.55	.51
PIPER$_{\text{SS}}$-Rel	.40	.42	.41	.47	.39	.43	.45	.52	.48	.52	.36	.42	.48	.55	.51
PIPER$_{\text{SS}}$-Sel	.40	.44	.42	.47	.40	.43	.45	.53	.48	.52	.39	.45	.45	.57	.50
PIPER$_{\text{AR}}$-Rel	.26	.38	.31	.46	.45	.46	.54	.45	.49	.49	.49	.49	.48	.53	.51
PIPER$_{\text{AR}}$-Sel	.26	.38	.31	.46	.45	.45	.54	.45	.49	.48	.50	.49	.48	.55	.51
AutoSlog-TS	.33	.49	.40	.52	.33	.41	.54	.59	.56	.48	.54	.51	.38	.45	.41
NB-Ex	.36	.34	.35	.35	.46	.40	.53	.49	.51	.50	.50	.50	1.00	.05	.10
Sem-Ex	.34	.46	.39	.30	.55	.39	.24	.78	.36	.18	.48	.26	.34	.69	.45

Disease Outbreaks (ProMed)						
	Disease			**Victim**		
Model	*P*	*R*	*F*	*P*	*R*	*F*
PIPER$_{\text{MS}}$-All	.32	.59	.41	.34	.53	.41
PIPER$_{\text{MS}}$-Rel	.37	.50	.43	.40	.45	.43
PIPER$_{\text{MS}}$-Sel	.35	.54	.43	.37	.52	.43
PIPER$_{\text{SS}}$-Rel	.33	.58	.42	.35	.53	.42
PIPER$_{\text{SS}}$-Sel	.33	.58	.42	.35	.53	.42
PIPER$_{\text{AR}}$-Rel	.43	.51	.47	.33	.43	.38
PIPER$_{\text{AR}}$-Sel	.40	.52	.45	.30	.45	.36
AutoSlog-TS	.33	.60	.43	.36	.49	.41
NB-Ex	.34	.59	.43	.47	.39	.43
Sem-Ex	.12	.70	.20	.12	.65	.20

also has the same pattern learning component (SemAff), its IE results are also quite similar. Overall, we see that PIPER$_{\text{MIL/SemAff}}$ has a slightly higher precision on all event roles except **PerpOrg** and **Target** indicating that it is slightly better at filtering out unwanted nonevent sentences. A comparison between PIPER$_{\text{MS}}$-Rel and PIPER$_{\text{MS}}$-Sel shows that the use of primary and secondary patterns with this sentential event recognizer results in strong recall gains on almost all event roles. This indicates that the PIPER$_{\text{MS}}$-Sel model is preferred over the other two PIPER$_{\text{MIL/SemAff}}$ models. A comparison across all the PIPER models shows that the supervised model PIPER$_{\text{Anskey/LexAff}}$ performs effective IE on most event roles except **PerpInd** in the terrorist events IE task, and **Victim** in the disease outbreaks IE task. The weakly supervised models for PIPER outperform the baselines on these event roles. Additionally, we observe that the weakly supervised model PIPER$_{\text{MIL/SemAff}}$ is fully automated and uses less supervision compared to the best (AutoSlog-TS) baseline, and yet matches its overall performance.

6.6 Unified Probabilistic Model Evaluation

The evaluation of the pipelined approach for IE showed that recognizing event descriptions in text allows the use of weaker contextual evidence for recognizing event role fillers. The weaker contextual evidence is represented as extraction patterns, which are learned

primarily on the basis of the semantics of their extractions. We find that for many event roles, this strategy outperforms existing approaches that rely only on the local context surrounding the candidate phrases. Our pipelined strategy is fully automated, and closely matches the overall performance of the AutoSlog-TS system, a pattern-based IE system benefitting from the oversight of a human expert.

This section presents an evaluation of GLACIER, a unified probabilistic model for IE that aims to achieve even better IE performance. The model uses a probabilistic sentential event recognizer, and a probabilistic plausible role-filler recognizer, both of which compute probabilities using several types of contextual clues seen in a given text. The two probabilities are combined into a single joint probability, based on which the extraction decisions are finally made. An evaluation of this model is presented here.

The implementation of the two components relies on machine learning classifiers, from which probabilities can be obtained. The sentential component, which was evaluated separately in Section 6.2, can be implemented as a Naïve Bayes classifier or as an SVM classifier with normalized decision scores. The plausible role-filler recognizer is implemented as Naïve Bayes classifier. These implementations of the two components lead to two primary configurations of GLACIER: (a) $\text{GLACIER}_{\text{NB/NB}}$, a model unifying probabilities from a Naïve Bayes sentential event recognizer and a Naïve Bayes plausible role-filler recognizer, and (b) $\text{GLACIER}_{\text{SVM/NB}}$, a model unifying probabilities from a normalized SVM sentential event recognizer and a Naïve Bayes plausible role-filler recognizer.

To perform this evaluation, the unified models were applied to a set of test documents, and a probability was generated for each noun phrase in the documents. This probability represents the confidence of the model in its assessment of an event role filler. We applied a threshold to this probability to determine whether to extract the phrase as a role filler of an event. To enable the extraction of fillers for several different event roles, a model was trained for each event role. The event roles extracted from the test documents were then compared against the answer strings listed in the answer key templates. Head noun scoring was used to determine if an extraction matched an answer string. Based on this, the precision, recall and F-score of the model were computed.

6.6.1 $\text{GLACIER}_{\text{NB/NB}}$

The $\text{GLACIER}_{\text{NB/NB}}$ model uses Naïve Bayes classifiers for its probability estimates. The sentential event recognizer is a Naïve Bayes classifier that analyzes the features appearing in a sentence, and computes a probability of the sentence being an event sentence, based on these features. Similarly, the plausible role-filler recognizer is also a Naïve Bayes classifier,

which uses the various features associated with a noun phrase to compute a probability of it being an event role filler. The primary variations in this model occur from the way it is trained.

The first model evaluated here consists of a sentential event recognizer and plausible role-filler recognizer both trained using approximate annotations from answer key templates. For the terrorist events, the two components were trained on approximate annotations applied to 1,300 training documents. For disease outbreaks, since the number of documents with answer key templates is much smaller, these models were trained using approximate annotations on 125 training documents. The Weka Toolkit [122] implementation of the Naïve Bayes with default configuration settings was used for both components of the model. An evaluation of this configuration on the two IE test sets is presented in the highlighted rows of Table 6.17. Cutoffs of 0.5, 0.7 and 0.9 are applied to the joint probability computed to make extraction decisions about noun phrases. The rows labeled "GLACIER$_{NN}$ 0.5" present the results for the 0.5 cutoff. Similarly, rows labeled "GLACIER$_{NN}$ 0.7" and "GLACIER$_{NN}$ 0.9" present results for the 0.7 and 0.9 cutoffs, respectively.

One of the striking observations within these results is the high recall obtained for each of the event roles. The table shows that the 0.9 cutoff clearly has better results across the board for all event roles as a result of the high recall. On the terrorist events data set, three event roles (**PerpInd**, **Victim** and **Weapon**) have F-scores over the best baseline

Table 6.17: GLACIER$_{NB/NB}$ evaluation on the test data

Terrorist Events (MUC-4)															
	PerpInd			**PerpOrg**			**Target**			**Victim**			**Weapon**		
Model	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
GLACIER$_{NN}$ 0.5	.27	.68	.39	.24	.56	.34	.28	.76	.41	.30	.76	.44	.36	.59	.45
GLACIER$_{NN}$ 0.7	.32	.64	.43	.26	.53	.35	.32	.76	.45	.39	.66	.49	.39	.57	.46
GLACIER$_{NN}$ 0.9	.39	.59	.47	.33	.51	.40	.39	.72	.51	.52	.54	.53	.47	.55	.51
PIPER$_{MS}$-Sel	.44	.41	.42	.47	.40	.43	.44	.51	.47	.55	.39	.45	.48	.55	.51
PIPER$_{SS}$-Sel	.40	.44	.42	.47	.40	.43	.45	.53	.48	.52	.39	.45	.45	.57	.50
PIPER$_{AR}$-Rel	.26	.38	.31	.46	.45	.46	.54	.45	.49	.49	.49	.49	.48	.53	.51
AutoSlog-TS	.33	.49	.40	.52	.33	.41	.54	.59	.56	.48	.54	.51	.38	.45	.41
NB-Ex	.36	.34	.35	.35	.46	.40	.53	.49	.51	.50	.50	.50	1.00	.05	.10
Sem-Ex	.34	.46	.39	.30	.55	.39	.24	.78	.36	.18	.48	.26	.34	.69	.45

Disease Outbreaks (ProMed)						
	Disease			**Victim**		
Model	*P*	*R*	*F*	*P*	*R*	*F*
GLACIER$_{NN}$ 0.5	.28	.75	.41	.22	.61	.33
GLACIER$_{NN}$ 0.7	.31	.70	.43	.24	.57	.34
GLACIER$_{NN}$ 0.9	.34	.59	.44	.32	.53	.40
PIPER$_{MS}$-Sel	.35	.54	.43	.37	.52	.43
PIPER$_{SS}$-Sel	.33	.58	.42	.35	.53	.42
PIPER$_{AR}$-Rel	.43	.51	.47	.33	.43	.38
AutoSlog-TS	.33	.60	.43	.36	.49	.41
NB-Ex	.34	.59	.43	.47	.39	.43
Sem-Ex	.12	.70	.20	.12	.65	.20

(by 7%, 2% and 10%, respectively) as a result of the high recall. The remaining two event roles, **PerpOrg** and **Target**, trail the best baseline by 1% and 5% respectively. The disease outbreaks domain does not see sharp performance gains like the terrorist events domain. The overall F-scores in this domain are slightly lower to those obtained by the PIPER model and the baseline IE systems. This behavior of the $\text{GLACIER}_{\text{NB/NB}}$ model on this domain is probably because of the small quantity of training data available in this domain.

The previous experiments showed that an effective IE system can be constructed even with smaller quantities of training data. Additionally, we note that all of these training data were in the form of approximate annotations with answer key templates. Since these approximate annotations were noisy, we conducted another experiment to see if a small amount of human annotated data could give us an improvement over the small amount of approximate annotations. Recall that we have human annotations for event sentences on a subset of 100 documents (TST3) in the MUC-4 terrorist events test data (of 200 documents: TST3+TST4). In the disease outbreaks domain, human annotations for event sentences are available for the entire 120 document test set. These human annotations were used in this experimental study.

Since all of the available human annotations are on the test data, a five-fold cross-validation was done across the test set. In the disease outbreaks domain, five splits or folds of the 120 document data set were created, with each fold containing 96 training documents and 24 test documents. In the terrorist events domain, the 100 annotated documents (TST3) were split into five folds, with each fold containing 80 documents for training. For testing each fold, however, the 20 test documents from the fold were combined with the remaining 100 documents (TST4), and the combined set of 120 documents was used for testing. The results were computed as average performance scores across the five folds in each domain.

To evaluate each fold, a Naïve Bayes sentential event recognizer was trained on the human annotations in the training data, and a plausible role-filler recognizer was a Naïve Bayes trained on approximate annotations from answer keys. The unified model composed of these two components was applied to the test documents in the fold, and the IE precision, recall and F-score were computed using the answer key templates associated with the test documents. An average of these scores was then computed across the five folds. Since the test documents in the five folds cover the entire official test sets, the performance numbers from this five-fold cross validation experiment are directly comparable to the previously obtained results.

The highlighted rows in Table 6.18 present the results of this experiment. From the

Table 6.18: GLACIER$_{NB/NB}$ five-fold cross validation using human annotations

Terrorist Events (MUC-4)															
	PerpInd			**PerpOrg**			**Target**			**Victim**			**Weapon**		
Model	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
GLACIER$_{NN}$-Hum	.41	.54	.47	.38	.18	.25	.39	.26	.31	.38	.28	.32	.96	.38	.54
GLACIER$_{NN}$ 0.9	.39	.59	.47	.33	.51	.40	.39	.72	.51	.52	.54	.53	.47	.55	.51
PIPER$_{MS}$-Sel	.44	.41	.42	.47	.40	.43	.44	.51	.47	.55	.39	.45	.48	.55	.51
PIPER$_{SS}$-Sel	.40	.44	.42	.47	.40	.43	.45	.53	.48	.52	.39	.45	.45	.57	.50
PIPER$_{AR}$-Rel	.26	.38	.31	.46	.45	.46	.54	.45	.49	.49	.49	.49	.48	.53	.51
AutoSlog-TS	.33	.49	.40	.52	.33	.41	.54	.59	.56	.48	.54	.51	.38	.45	.41
NB-Ex	.36	.34	.35	.35	.46	.40	.53	.49	.51	.50	.50	.50	1.00	.05	.10
Sem-Ex	.34	.46	.39	.30	.55	.39	.24	.78	.36	.18	.48	.26	.34	.69	.45

Disease Outbreaks (ProMed)						
	Disease			**Victim**		
Model	*P*	*R*	*F*	*P*	*R*	*F*
GLACIER$_{NN}$-Hum	.41	.61	.49	.38	.52	.44
GLACIER$_{NN}$ 0.9	.34	.59	.44	.32	.53	.40
PIPER$_{MS}$-Sel	.35	.54	.43	.37	.52	.43
PIPER$_{SS}$-Sel	.33	.58	.42	.35	.53	.42
PIPER$_{AR}$-Rel	.43	.51	.47	.33	.43	.38
AutoSlog-TS	.33	.60	.43	.36	.49	.41
NB-Ex	.34	.59	.43	.47	.39	.43
Sem-Ex	.12	.70	.20	.12	.65	.20

performance numbers we find that, for terrorist events, 80 human-annotated documents are not as effective as 1300 documents with approximate annotations. In the disease outbreaks task, on the other hand, we see good IE performance on both event roles, which suggests that human annotations (even in small quantities) are useful for accurately identifying event sentences in this domain. Overall, this helps raise the precision on both the disease outbreaks event roles, while maintaining a good recall. Compared to the GLACIER$_{NB/NB}$ model trained with approximate annotations, this model achieves 5% and 4% higher F-scores on **Disease** and **Victim** event roles respectively. The higher F-scores are because of higher IE precision resulting from a more accurate sentential event recognizer. Furthermore, these F-scores on the disease outbreaks IE task are the highest compared to the baselines and the other PIPER and GLACIER models.

6.6.2 GLACIER$_{SVM/NB}$

It has been observed that Naïve Bayes classifiers tend to overestimate the probability of one class over the remaining classes [33, 127, 68]. As a result of this, the generated probability values many times do not reflect the true underlying distribution of the classes. This issue particularly affects the sentential event recognizer in our model, more so than it does the plausible role-filler recognizer. Thus, here we incorporate an alternate strategy for obtaining sentential probabilities, with the use of an SVM classifier. The probabilities generated are not true probabilities, but an approximation based on the normalization of the

scores generated by an SVM. An evaluation of this model (GLACIER$_{SVM/NB}$) is presented here.

The IE model evaluated here consists of a sentential event recognizer based on the normalized SVM scores, and a plausible role-filler recognizer based on a Naïve Bayes classifier. Both classifiers are trained using approximate annotations from answer key templates. For the terrorist events, the two components were trained on approximate annotations in 1,300 MUC-4 training documents. For disease outbreaks, since the number of documents with answer key templates is much smaller, these models were trained on approximate annotations in 125 ProMed documents. An evaluation of this configuration on the two IE test sets is presented in the highlighted rows of Table 6.19. Unlike the GLACIER$_{NB/NB}$ model, which achieves good performance with a 0.9 cutoff on the joint probability, this model does better with the more natural cutoff of 0.5. Additionally, experiments showed that a 0.4 cutoff also works well. IE results corresponding to cutoffs of 0.4, 0.5 and 0.6 applied to the joint probability are shown in the table. The rows labeled "GLACIER$_{SN}$ 0.4" present the results for the 0.4 cutoff. Similarly, rows labeled "GLACIER$_{SN}$ 0.5" and "GLACIER$_{SN}$ 0.6" present results for the 0.5 and 0.6 cutoffs, respectively. The 0.9 cutoff does not work well at all in this model, indicating that the probabilities are more

Table 6.19: GLACIER$_{SVM/NB}$ evaluation on the test data

Terrorist Events (MUC-4)															
	PerpInd			**PerpOrg**			**Target**			**Victim**			**Weapon**		
Model	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
GLACIER$_{SN}$ 0.4	.51	.58	.54	.34	.45	.38	.42	.72	.53	.55	.58	.56	.57	.53	.55
GLACIER$_{SN}$ 0.5	.66	.47	.55	.41	.26	.32	.50	.62	.55	.62	.36	.45	.64	.43	.52
GLACIER$_{SN}$ 0.6	.73	.21	.32	.39	.11	.17	.58	.39	.47	.64	.20	.30	.67	.36	.47
GLACIER$_{NN}$-Hum	.41	.54	.47	.38	.18	.25	.39	.26	.31	.38	.28	.32	.96	.38	.54
GLACIER$_{NN}$ 0.9	.39	.59	.47	.33	.51	.40	.39	.72	.51	.52	.54	.53	.47	.55	.51
PIPER$_{MS}$-Sel	.44	.41	.42	.47	.40	.43	.44	.51	.47	.55	.39	.45	.48	.55	.51
PIPER$_{SS}$-Sel	.40	.44	.42	.47	.40	.43	.45	.53	.48	.52	.39	.45	.45	.57	.50
PIPER$_{AR}$-Rel	.26	.38	.31	.46	.45	.46	.54	.45	.49	.49	.49	.49	.48	.53	.51
AutoSlog-TS	.33	.49	.40	.52	.33	.41	.54	.59	.56	.48	.54	.51	.38	.45	.41
NB-Ex	.36	.34	.35	.35	.46	.40	.53	.49	.51	.50	.50	.50	1.00	.05	.10
Sem-Ex	.34	.46	.39	.30	.55	.39	.24	.78	.36	.18	.48	.26	.34	.69	.45

Disease Outbreaks (ProMed)						
	Disease			**Victim**		
Model	*P*	*R*	*F*	*P*	*R*	*F*
GLACIER$_{SN}$ 0.4	.32	.70	.44	.29	.55	.38
GLACIER$_{SN}$ 0.5	.39	.55	.46	.35	.39	.37
GLACIER$_{SN}$ 0.6	.42	.41	.41	.45	.30	.36
GLACIER$_{NN}$-Hum	.41	.61	.49	.38	.52	.44
GLACIER$_{NN}$ 0.9	.34	.59	.44	.32	.53	.40
PIPER$_{MS}$-Sel	.35	.54	.43	.37	.52	.43
PIPER$_{SS}$-Sel	.33	.58	.42	.35	.53	.42
PIPER$_{AR}$-Rel	.43	.51	.47	.33	.43	.38
AutoSlog-TS	.33	.60	.43	.36	.49	.41
NB-Ex	.34	.59	.43	.47	.39	.43
Sem-Ex	.12	.70	.20	.12	.65	.20

evenly distributed across the 0.0 to 0.1 range with the use of an SVM.

Observe that the probabilities obtained from the SVM greatly improve the performance of the model. The overall F-score in the terrorism domain surpasses that of the GLACIER$_{NB/NB}$ model. The F-scores of all the individual event roles are higher (except **PerpOrg**, which is 3% lower) than those obtained with GLACIER$_{NB/NB}$. Compared with the best PIPER model, we see gains of 12%, 5%, 11% and 5% on the **PerpInd**, **Target**, **Victim** and **Weapon** F-scores, respectively. Only the **PerpOrg** event role has a 5% lower F-score. Compared with the best (AutoSlog-TS) baseline, we see F-score gains of 14%, 5% and 14% on **PerpInd**, **Victim** and **Weapon** event roles, respectively. The F-scores on **PerpOrg** and **Target** event roles are both 3% lower than AutoSlog-TS. Thus, for the terrorist events task we see performance gains with this model over the baselines and the other IE models. For the disease outbreaks task, the **Disease** event role does well with this model, but the performance on the **Victim** role is 4% lower than AutoSlog-TS. Again, the limited quantity of training data in this domain is the likely cause for this.

6.7 Statistical Significance

This research has shown several variations of a pipelined and a unified probabilistic model for IE. The experiments presented in the previous section have shown that these models perform as well as or better than the baseline systems on the test sets. In this section, we run some statistical significance tests to determine which of the PIPER and GLACIER models are statistically significantly better than the baseline systems.

For conducting the statistical significance tests we use a bootstrap resampling method described by Koehn [58] for Machine Translation. We adapt his methodology to our task of event-based IE. In this method, a large number of sample data sets are randomly drawn (with replacement) from a test data set. To determine if an IE model is statistically significantly better than another IE model, the two models are applied to each of the randomly sampled data sets and the difference in their performance is measured. If the first IE model is found to have a better performance than the second IE model on over 95% of the random samples, then the first IE model is considered to be better than the second IE model with 95% statistical significance.

Using this methodology, we compare each of the PIPER and GLACIER models with each of the three baseline IE systems. In each case, the statistical significance is measured over the aggregate F-score of the models on all event roles. Table 6.20 summarizes the results of these tests for terrorist events. The random sampling of the test data is performed over the terrorist events test data. The PIPER and GLACIER models listed in the rows are compared

Table 6.20: Statistical significance tests for overall performance on terrorist events

	AutoSlog-TS	NB-Ex	Sem-Ex
GLACIER$_{SN}$ 0.4	95%	99%	99%
GLACIER$_{NN}$ 0.9	-	95%	99%
PIPER$_{MS}$-Sel	-	-	99%
PIPER$_{SS}$-Sel	-	-	99%
PIPER$_{AR}$-Rel	-	-	99%

against the baseline IE systems listed in the columns. If the model listed in the row is not found to be statistically significantly better than the baseline listed in the column, then the corresponding cell in this table is empty. Otherwise, the cell contains the confidence level of the statistical significance.

From the table we see that the GLACIER$_{SVM/NB}$ is better than all three baselines with at least 95% statistical significance (99% in two cases). GLACIER$_{NB/NB}$ is found to perform better than two of the baseline systems. The PIPER models are found to be not significantly better than the AutoSlog-TS and the Unconditioned Naïve Bayes (NB-Ex) baselines, but are clearly superior to the Semantic Class Extractor (Sem-Ex) baseline. Table 6.21 shows a breakdown of the statistical significance tests for each of the event roles in the terrorist events IE task.

For disease outbreaks, we found that compared to AutoSlog-TS and the Unconditioned Naïve Bayes baselines, none of the PIPER or GLACIER models were statistically significantly better than the baselines. On the other hand, compared to the Semantic Class Extractor baseline, all of the PIPER and GLACIER models were found to be statistically significantly

Table 6.21: Statistical significance tests for event roles in terrorist events

	PerpInd	**PerpOrg**	**Target**	**Victim**	**Weapon**
Baseline: AutoSlog-TS					
GLACIER$_{SN}$ 0.4	99%	-	-	-	95%
GLACIER$_{NN}$ 0.9	-	-	-	-	-
PIPER$_{MS}$-Sel	-	-	-	-	90%
PIPER$_{SS}$-Sel	-	-	-	-	-
PIPER$_{AR}$-Rel	-	-	-	-	-
Baseline: NB-Ex					
GLACIER$_{SN}$ 0.4	99%	-	-	90%	99%
GLACIER$_{NN}$ 0.9	95%	-	-	-	99%
PIPER$_{MS}$-Sel	-	-	-	-	99%
PIPER$_{SS}$-Sel	-	-	-	-	99%
PIPER$_{AR}$-Rel	-	-	-	-	99%
Baseline: Sem-Ex					
GLACIER$_{SN}$ 0.4	99%	-	99%	99%	-
GLACIER$_{NN}$ 0.9	90%	-	99%	99%	-
PIPER$_{MS}$-Sel	-	-	95%	99%	-
PIPER$_{SS}$-Sel	-	-	95%	99%	-
PIPER$_{AR}$-Rel	-	-	95%	99%	-

better. The reason for this outcome is most likely the result of the small quantity of training data available in this domain.

6.8 Learning Curves

In this research, we found that the unified probabilistic model (GLACIER) is an effective model for extracting event role fillers of disease outbreaks events, in spite of the small training data set used. We would now like to see if we can get away with using smaller amounts of training data for terrorist events as well. Thus, this next experiment studies the effect of varying training data sizes on IE performance in the terrorism domain. The $\text{GLACIER}_{\text{NB/NB}}$ (0.9) and $\text{GLACIER}_{\text{SVM/NB}}$ (0.4) models are trained with randomly selected subsets of the 1,300 training documents about terrorist events. The changes in performance for varying quantities of training data are recorded, and are used to generate learning curves for this data set.

Table 6.22 shows IE results with models trained using smaller quantities of training data. The columns labeled "Train100," "Train200" and "Train300" contain the IE performance scores corresponding to training data set sizes of 100, 200 and 300 documents, respectively. In the $\text{GLACIER}_{\text{NB/NB}}$ model, observe that the F-score rapidly ramps up from 0.36 to 0.46 as the amount of corresponding training data increases from 100 to 300. This performance is just 3% short of the 0.49 F-score achieved with 1,300 documents of training data. Thus, much smaller quantities of training data can be useful in this model for IE. The effect of the size of the training data set, however, varies across the various event roles in this IE task. The PerpOrg and Weapon event roles are largely unaffected by the small quantity of

Table 6.22: GLACIER evaluation with reduced training

Terrorist Events (MUC-4)									
	Train100			Train200			Train300		
Event Role	*Pr*	*Rec*	*F*	*Pr*	*Rec*	*F*	*Pr*	*Rec*	*F*
$\text{GLACIER}_{\text{NB/NB}}$									
PerpInd	0.61	0.20	0.30	0.39	0.47	0.42	0.36	0.53	0.43
PerpOrg	0.33	0.41	0.37	0.35	0.45	0.40	0.33	0.41	0.37
Target	0.38	0.14	0.20	0.45	0.45	0.45	0.39	0.52	0.44
Victim	0.61	0.20	0.30	0.51	0.33	0.40	0.51	0.39	0.44
Weapon	0.85	0.38	0.52	0.69	0.41	0.52	0.73	0.41	0.53
Aggregate	0.56	0.26	0.36	0.48	0.42	0.45	0.46	0.45	0.46
$\text{GLACIER}_{\text{SVM/NB}}$									
PerpInd	0.42	0.31	0.35	0.37	0.50	0.42	0.37	0.52	0.43
PerpOrg	0.29	0.47	0.36	0.32	0.48	0.39	0.36	0.43	0.39
Target	0.48	0.39	0.43	0.43	0.50	0.47	0.46	0.61	0.52
Victim	0.40	0.31	0.35	0.44	0.43	0.44	0.42	0.45	0.44
Weapon	0.83	0.41	0.55	0.65	0.45	0.53	0.70	0.45	0.55
Aggregate	0.48	0.38	0.42	0.44	0.47	0.46	0.46	0.49	0.48

training data and even with 100 documents for training on these two event roles, the model manages to achieve close to its F-score with 1,300 training documents. In the remaining three event roles, we see a much stronger effect of the size of the training data set. In all three cases we see increases in F-scores of 13% or more, each accompanied with sharp increases in recall.

We have a similar story with GLACIER$_{\text{SVM/NB}}$. Here too performance rises with more training data. The overall F-score rises from 0.42 to 0.48 by increasing the training data set size from 100 to 300 documents. Within the breakdown of the scores per event role, we observe that the Weapon event role has a relatively constant F-score, but has a sharp drop in precision from 0.83 to 0.70, for a corresponding increase in recall from 0.41 to 0.45. Likewise, the PerpOrg event role's F-score remains relatively constant, but its precision increases from 0.29 to 0.36 for a corresponding recall drop from 0.47 to 0.43. The F-scores of both event roles are close to those achieved with the full 1,300 training documents. All of the remaining event roles are strongly affected by the small training set of 100 documents, but improve in performance with the 200 and 300 document training sets.

Figure 6.2 and Figure 6.3 present learning curves for the GLACIER$_{\text{NB/NB}}$ model and the

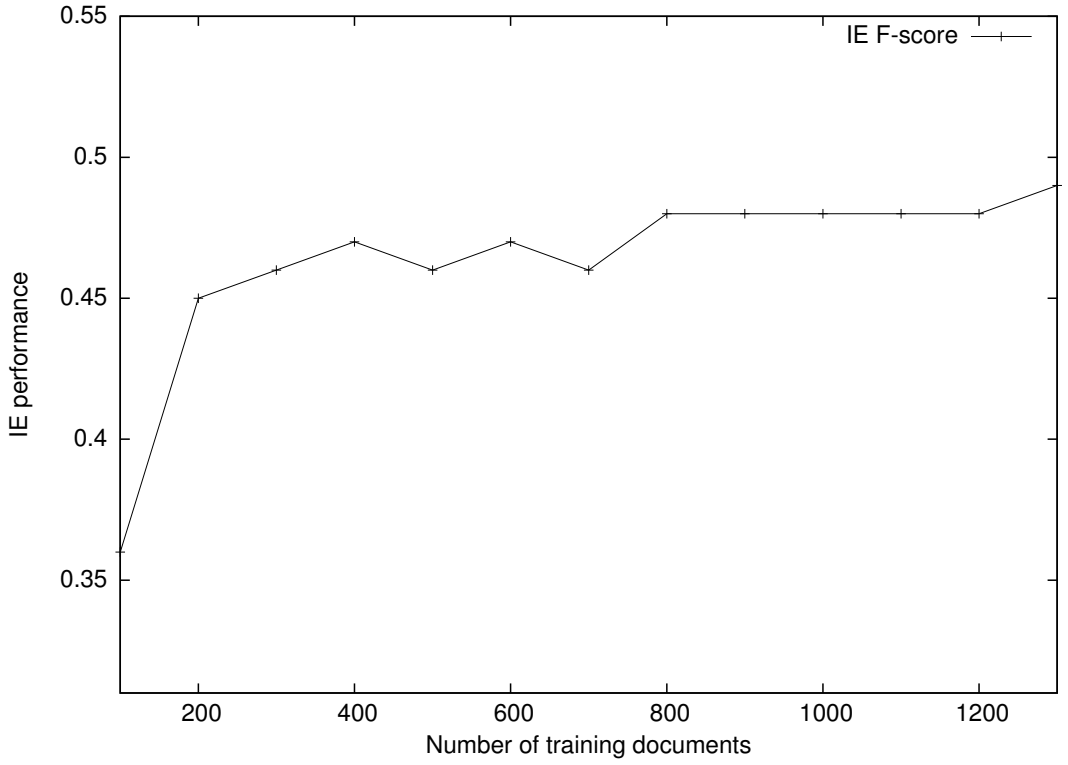


Figure 6.2: GLACIER$_{\text{NB/NB}}$ learning curve

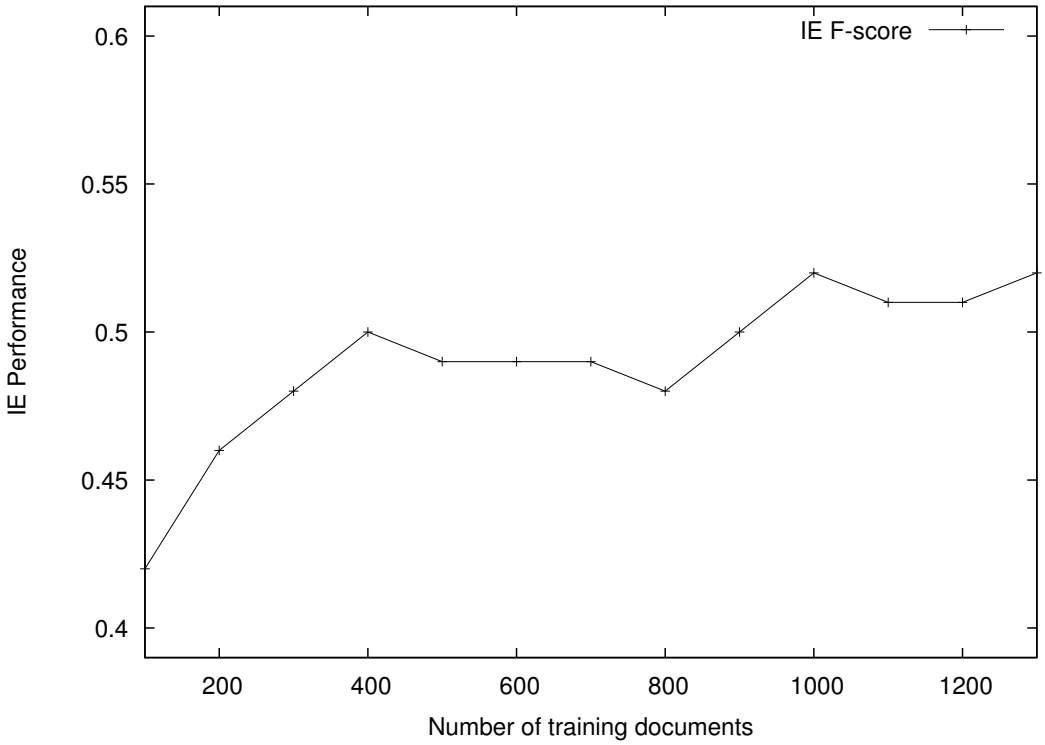


Figure 6.3: GLACIER$_{\text{SVM/NB}}$ learning curve

GLACIER$_{\text{SVM/NB}}$ model, respectively. The graphs plot IE performance (F-score) against training data set size for terrorist events. In GLACIER$_{\text{NB/NB}}$ we see that the F-score rapidly increases to a value of 0.45 with increasing training data up to 200 documents. After this the F-score remains relatively stable as more training documents are added. Beyond 200 documents, the IE recall continues to increase gradually, but at the cost of IE precision. Similarly, in GLACIER$_{\text{SVM/NB}}$, the graph shows that the F-score rapidly increases to an F-score of almost 0.50 with increasing training data up to 400 documents. Beyond this the F-score increases gradually.

Both graphs illustrate that even with smaller quantities of training data for terrorist events, the GLACIER models achieve close to their best performance. From this we could surmise that the changes in performance with varying quantities of training data are possibly replicated in other domains too. Thus, we further conjecture that additional training data could benefit the models for disease outbreak events as well.

6.9 Feature Analysis

Many different types of features are employed in the sentential event recognizers and the plausible role filler recognizers. In this section we study the contribution of the various

types of features in the performance of the model component. To measure the effect of a feature type, the overall IE performance of the model is measured first with the feature type included in the model and then with the feature type excluded from the model. The difference in performance across the two configurations can indicate the usefulness of the feature in that model. Using this strategy we can measure the effect of various feature types employed in the models.

Since the GLACIER$_{\text{SVM/NB}}$ and GLACIER$_{\text{NB/NB}}$ are the best performing models for terrorist events and disease outbreak events, respectively, we use these as our reference models for this feature analysis study. Each of these models contains a sentential event recognizer and a plausible role-filler recognizer, which use contextual features for generating probability estimates. Recall from Chapter 4 that the plausible role-filler recognizer uses five types of phrase-level features: *lexical heads*, *semantic classes*, *lexico-syntactic patterns*, *phrase characteristics* and *named entities*. Similarly, the sentential event recognizer contains all of these phrase-level feature types and three additional sentence-level feature types: *sentence length*, *bag of words* and *verb tense*. Each of these feature types are analyzed here. Table 6.23 and Table 6.24 present the results of these analysis experiments, and a description of these experiments follows in this section. The first row labeled "All Features" in each of the tables contains the performance scores of the model using all of the features. The remaining rows in the table contain the model performance for various configurations of the features used.

Since all of the phrase-level features used in the plausible role-filler recognizer are also used by the sentential event recognizer, there is significant overlap in the feature sets of the two components of the IE model. Our first analysis experiment aims to determine if segregating this feature set by using only the phrase-level features in the plausible role-filler recognizer and using only the sentence-level features in the sentential event recognizer can enable the classifiers to make better decisions. The rows labeled "Segregated" in each of the

Table 6.23: Feature analysis of GLACIER$_{\text{SVM/NB}}$ for terrorist events

	PerpInd			**PerpOrg**			**Target**			**Victim**			**Weapon**		
Model	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
All Features	.51	.58	.54	.34	.45	.38	.42	.72	.53	.55	.58	.56	.57	.53	.55
Segregated	.51	.60	.55	.32	.48	.39	.40	.75	.52	.51	.59	.54	.55	.52	.53
Phrase-head	.55	.53	.54	.32	.45	.37	.43	.68	.53	.54	.52	.53	.55	.52	.53
Phrase-sem	.70	.41	.51	.70	.23	.35	.83	.39	.53	.68	.33	.44	.74	.30	.43
Phrase-pattern	.47	.60	.53	.31	.49	.38	.37	.73	.49	.45	.52	.48	.57	.60	.59
Phrase-char	.53	.58	.55	.32	.47	.38	.40	.75	.52	.49	.57	.53	.56	.52	.54
Phrase-ner	.51	.60	.55	.33	.48	.39	.40	.74	.52	.49	.56	.52	.55	.52	.53
Sentence-length	.50	.60	.55	.33	.48	.39	.40	.75	.52	.50	.59	.54	.55	.52	.53
Sentence-bow	.16	.70	.27	.10	.69	.17	.18	.82	.29	.14	.83	.24	.25	.66	.36
Sentence-tense	.50	.60	.55	.32	.48	.39	.40	.74	.52	.51	.59	.55	.56	.52	.54

Table 6.24: Feature analysis of GLACIER$_{NB/NB}$ for disease outbreaks

	Disease			Victim		
Model	*P*	*R*	*F*	*P*	*R*	*F*
All Features	.34	.59	.44	.32	.53	.40
Segregated	.38	.53	.44	.38	.48	.43
Phrasal-head	.43	.50	.46	.41	.46	.43
Phrasal-sem	.63	.30	.41	.63	.30	.40
Phrasal-pattern	.38	.53	.44	.34	.49	.40
Phrasal-char	.36	.51	.43	.45	.44	.44
Phrasal-ner	.37	.55	.45	.39	.48	.43
Sentence-length	.39	.51	.44	.38	.48	.42
Sentence-bow	.00	.00	.00	.00	.00	.00
Sentence-tense	.39	.53	.45	.38	.47	.42

tables presents the results of having no overlapping features in the two components of the models. With this modification we see overall performance improvement on **PerpInd** and **PerpOrg** event roles in terrorist events, and on the **Victim** event role in disease outbreaks. On the remaining event roles, however, the performance is slightly lower in terrorist events, and no F-score change in the **Disease** event role in disease outbreaks. On the whole, overlapping features do not have a major impact on performance.

The remaining experiments presented in the two tables take the "Segregated" model and discard a subset of the features from the model and measure the impact on their performance. The rows labeled "Phrase-*" present the effect of discarding phrase-level features (used in the plausible role-filler recognizer), and the rows labeled "Sentence-*" present the effect of discarding the sentence-level features (used by the sentential event recognizer). In the case of phrase-level features, features of each of the types — *lexical heads*, *semantic classes*, *lexico-syntactic patterns*, *phrase characteristics* and *named entities* — are discarded in turn, and the measured performance of the model is presented, respectively, in the rows labeled "Phrase-head," "Phrase-sem," "Phrase-pattern," "Phrase-char" and "Phrase-ner." Similarly, discarding each of the sentence-level features — *sentence length*, *bag of words* and *verb tense* — in turn, generates the IE results presented, respectively, in the rows labeled "Sentence-length," "Sentence-bow" and "Sentence-tense."

Observe that the sentence-level *bag of words* features have the maximum impact on performance of the models. In fact, in both event types we find that after removing the *bag of words* features, the sentential event recognizer was unable to recognize event sentences. In the case of terrorist events the SVM assigns a probability of 1.0 to all sentences, while in the disease outbreak events the Naïve Bayes classifier assigns a probability of 0.0 to all sentences. Needless to say, this drastically affects overall IE performance.

While, in most of the remaining feature study experiments we see small changes in the overall performance of the models, we do see a strong impact of some of the features.

Of particular note is the effect of the phrase-level *semantic classes* features. In both event types, removing the *semantic classes features* results in a sharp drop in recall and a large improvement in precision. The **Victim** and **Weapon** event-roles in terrorist events are the most affected by the feature type, causing a drop in their F-score of almost 10%. The *semantic classes* features enable the model to generalize the phrasal contextual representation, which allows the system to gain recall points. However, the use of these features negatively impacts precision because it may be overgeneralizing the contextual representation in some cases.

For terrorist events, the *lexico-syntactic patterns* features are important for effectively identifying **PerpInd**, **Target** and **Victim** event roles. On the other hand, these features have a negative impact on the **Weapon** event role. Note that the event roles which benefit from the *lexico-syntactic patterns* features correspond exactly to those role filler strings whose lexical semantics are weak indicators of an event role. For example, the victim of a terrorist event can be any human mentioned in a text, and can be easily confused with other humans referenced in the text. Thus, the local context (represented by lexico-syntactic patterns) is necessary to resolve such confusions. The weapon used in the terrorist event, on the other hand, cannot easily be confused with other objects mentioned in the text. Thus, in many cases, just the semantics of a potential role filler string can be sufficient for identifying weapons of terrorist events. All of this suggests that it may be possible to identify beforehand the event roles that could potentially benefit from the *lexico-syntactic patterns* features and those that could benefit from the *semantic classes* features.

Of the remaining feature types, the phrase-level *lexical head* features in disease outbreak event tend to have a negative impact on the precision of **Disease** and **Victim** event roles. We observe that removing this feature results in an overall improvement in the F-score of the **Disease** event role.

6.10 Extractions with Weak Local Evidence

This dissertation has demonstrated the usefulness of information in the wider context exploited by the Piper and Glacier models for event-based IE. However, most of the empirical evidence presented in support of these models is in terms of overall IE performance. The recall gains achieved by these models illustrate the increased coverage that can be attributed to wider contextual information. Some of these gains can very well be a result of a smarter or better use of local information by these models. To clearly demonstrate the benefit of sentential event recognition, this section presents further analysis of the Glacier

model pertaining to its extraction decisions in the face of weak local contexts.

Our goal is to determine how much of the recall increase achieved by our unified model was in examples with weak local evidence. To keep this analysis more manageable, we focus on the GLACIER$_{\text{SVM/NB}}$ model used for extracting terrorist event information. This model is compared with the pattern-based AutoSlog-TS baseline system, and the classifier-based Unconditioned Naïve Bayes (NB-Ex) baseline system.

We find that on the test data, GLACIER$_{\text{SVM/NB}}$ achieves its recall gains by correctly extracting 144 additional role filler strings compared to the NB-Ex system. Compared to the AutoSlog-TS system, GLACIER$_{\text{SVM/NB}}$ is able to find 123 new role filler strings that had been missed by AutoSlog-TS. Taking a union of the extractions from the two baseline systems, we find that GLACIER$_{\text{SVM/NB}}$ still correctly found 79 additional role filler strings that were missed by both AutoSlog-TS and NB-Ex.

Since, AutoSlog-TS and NB-Ex both rely solely on the local context to determine the role filler extractions, we can surmise that the 79 new extractions by GLACIER$_{\text{SVM/NB}}$ result from its exploitation of weaker local contexts or from a smarter use of stronger local contexts. To determine the effect of sentential event recognition, we manually analyze the 79 additional extractions by GLACIER$_{\text{SVM/NB}}$, and count the number of examples that require the use of the sentential event recognizer in order to correctly extract role filler information. Specifically, we look at the immediate context of the extractions and determine if an extraction pattern or a local contextual feature can potentially enable the baseline AutoSlog-TS or NB-Ex systems to correctly spot the event role fillers. Our analysis indicates that 32 of the 79 additional extractions have no such evidence in the local context, and thus illustrate the benefits of sentential event recognition. The 32 examples identified in this analysis are listed in Appendix D. This analysis indicates that at least 41% of the recall improvement achieved by the unified model can be attributed to its exploitation of weaker local contexts with the use of sentential event recognition.

6.11 Error Analysis

As we have seen from the performance evaluations in the earlier sections, event-based IE is a difficult task. Even our best model does not achieve perfect precision or recall over the test data. As a result of the characteristics of the data, and the features and learning strategies employed, our models make some errors in correctly identifying event role filler strings for the events of interest. In contrast to the previous section, which examined the strengths of our IE models, this section presents an analysis of its errors on the test data.

In general, we find two types of errors in the use of our IE models. First, they fail to extract event role fillers that should have been extracted (false negatives). Second, they erroneously identify incorrect text spans as event role fillers (false positives). Both of these types of errors negatively impact overall performance of the IE model. In this analysis, we examine a random sample of sentences over which our models exhibit the two types of errors, and we present a discussion of the underlying data and model characteristics that result in these errors. Such an analysis could be used to help direct future research for further improvement in event-based IE performance.

We perform this analysis on our GLACIER$_{\text{SVM/NB}}$ model used for extracting terrorist event information. In our previous experiments we found that, on the test data, the precision of the model was impacted by the erroneous extraction of 386 incorrect text spans as event role fillers. Similarly, its recall was impacted by failing to extract 243 text spans as event role fillers mentioned in the answer keys. We randomly sample 30 errors of each type, and analyze the individual instances to determine the cause for these failures. In all of these cases we find that the causes for the errors can be broadly attributed to the two components of the model or to other related factors. Table 6.25 and Table 6.26 briefly summarize the results of this analysis. The percentages in each table may add up to more than 100%, because some of the errors were attributed to multiple reasons.

Among the 30 cases where the model erroneously extracts incorrect text spans as role fillers (false positives), we find that almost 15 (or 50%) of the cases can be attributed to weaknesses in the local extraction component (the *plausible role-filler recognizer* in the GLACIER$_{\text{SVM/NB}}$ model). In most of these (10 cases), the semantic class of the candidate test span strongly favors the extraction in an event context, and there is little other local contextual evidence to say otherwise. This typically happens in the case of *victim* and *target* event roles, whose semantics are easy to confuse with other nonvictim or nontarget entities mentioned in text. In about four cases the incorrectly extracted victims are, in fact, references to people in the text, who are describing or alluding to the relevant event.

Table 6.25: Error analysis of false positives

Reasons for false positives	
50%	Weaknesses in plausible role-filler recognizer
23%	Weaknesses in sentential event recognizer
17%	Nuances of MUC-4 extraction task
10%	No coreference resolution
10%	Parse errors

Table 6.26: Error analysis of false negatives

Reasons for false negatives	
47%	Weaknesses in plausible role-filler recognizer
27%	Weaknesses in sentential event recognizer
17%	Noun phrase extraction
10%	Parse errors
3%	No coreference resolution

Weaknesses in the sentential event recognition component has an impact on about seven (or 23%) of the erroneous extractions. In all of these cases, a nonevent sentence is incorrectly assigned a high probability of being an event sentence by the sentential event recognizer. The remaining errors are attributed to "other" factors related to the given IE task and the expressiveness of our model. For instance, in about five cases (or 17%) the extracted strings are deemed incorrect as a result of the finer nuances in the definition of terrorist events, which our model currently is unable to definitively capture. One such finer detail in the given terrorist event definition, for example, is the exclusion of violent attacks on military personnel as terrorist events. In these cases, the extractions by our model negatively impact our precision. Another cause for failure in three (or 10%) of the cases is due to the need for coreference resolution. In these three cases the extracted strings, in fact, corefer to valid answers, but are themselves semantically too general (e.g., "the perpetrators") to be considered a correct answer. Of all of the above cases, about three (or 10%) are the result of parse errors.

In the second part of this error analysis, we look at 30 randomly sampled cases where the model fails to extract an event role filler mentioned in the answer key (false negatives). About 14 (or 47%) of these failures result from insufficient evidence in the local context indicating a role filler. Of these 14, about five are cases where the semantics of the phrases are not clearly indicative of a role filler. In some of cases, the semantics of the phrases are misleading or ambiguous (e.g., "guerillas" as victims of terrorist events and "mine" as the weapon used). Among these are also cases where the model lacks mechanims to make inferences from the presented information. For example, the model correctly identifies the "residence of legislative assembly president" as the target of a terrorist event, but is unable to infer that the "legislative assembly president" is the victim of that event. Some failures are also the result of shortcomings of the sentential event recognizer. About eight (or 27%) cases result from the sentential event recognizer incorrectly assigning a low event sentence probability to sentences describing relevant events. There are also four cases where the role

fillers to be extracted appear in sentences that are not, in fact, event sentences. For our model to extract such role fillers would require the expansion of region size to include such sentences linked through discourse to event sentences. About five (or 17%) role fillers are missed by the model because of its focus on noun phrases. Our model considers each noun phrase in a sentence as a candidate for extraction. However, in these five cases, the role fillers appear as modifiers of the head noun within the noun phrases. Of all of the above missed extractions, about three (or 10%) are caused due to an incorrect parse from our shallow parser, and one (or 3%) is due to the lack of a coreference resolver in our model.

As seen from the above error analysis, a majority of the errors from our model can be overcome by strengthening our localized text extraction component. Other causes for the errors include the need for more accurate parsing, a coreference resolver, and expanding the scope of our model beyond just noun phrases.

6.12 Evaluation Summary

This chapter presented an extensive evaluation of several IE models. Before concluding this chapter, let us briefly recap the highlights of the various IE models that were presented and see how they compare against one another. The aggregate IE performance scores of these models is summarized in Table 6.27. Three baseline systems were used in this comparison. All of these baselines rely solely on the local context in making their extraction decisions. They also vary in the extent of supervision used to train the system. The first baseline is AutoSlog-TS, a pattern-based IE system that uses relevant and irrelevant documents, and some human expert oversight for pattern review. An unconditioned Naïve Bayes classifier (unconditioned-NB) that uses local contextual evidence to extract noun phrases is our second baseline. Finally, a semantic class extractor applied to event sentences identified by a self-trained SVM classifier forms the third baseline. These are compared against

Table 6.27: Summary of evaluation of IE models

	Terrorist Events			***Disease Outbreaks***		
Event Role	*Pr*	*Rec*	*F*	*Pr*	*Rec*	*F*
AutoSlog-TS	0.45	0.50	0.47	0.35	0.53	0.42
Unconstrained-NB	0.55	0.37	0.44	0.41	0.49	0.44
SemClass Extractor	0.24	0.57	0.34	0.12	0.68	0.20
PIPER$_{\text{Anskey/LexAff}}$	0.44	0.47	0.45	0.38	0.47	0.42
PIPER$_{\text{Self/SemAff}}$	0.46	0.47	0.46	0.34	0.56	0.42
PIPER$_{\text{MIL/SemAff}}$	0.48	0.45	0.46	0.36	0.53	0.43
GLACIER$_{\text{NB/NB}}$	0.42	0.58	0.49	0.33	0.56	0.42
GLACIER$_{\text{NB/NB}}$-Hum	0.50	0.32	0.39	0.40	0.57	0.47
GLACIER$_{\text{SVM/NB}}$	0.48	0.57	0.52	0.37	0.47	0.42

variations of the PIPER and GLACIER models. For the PIPER model, we have three variations — $\text{PIPER}_{\text{Anskey/LexAff}}$, $\text{PIPER}_{\text{Self/SemAff}}$ and $\text{PIPER}_{\text{MIL/SemAff}}$. Similarly, for the GLACIER model we have two variations — $\text{GLACIER}_{\text{NB/NB}}$ and $\text{GLACIER}_{\text{SVM/NB}}$. Furthermore, results from the use of a small quantity of additional manual sentence-level annotations for training $\text{GLACIER}_{\text{NB/NB}}$ are also presented in the row labeled $\text{GLACIER}_{\text{NB/NB}}$-Hum.

The results show that GLACIER models trained on the approximate annotations clearly surpass the baseline IE models. $\text{GLACIER}_{\text{SVM/NB}}$ achieves an F-score of 0.52, which is about 5% greater than the pattern-based IE system AutoSlog-TS. On the disease outbreaks data set, both the PIPER model and the GLACIER model match the performance of most baseline systems. The small amount of training data in this domain makes it difficult for the IE system to achieve higher performance. However, adding additional sentence-level annotations to this small data set improves the performance of $\text{GLACIER}_{\text{NB/NB}}$ to surpass that of the baseline systems. Table 6.28 shows the performance of the best models for the two event types, in comparison with the baseline model performance. Overall, we see that the unified probabilistic approach performs the best of all the approaches. Section 6.7 shows that the improvement in performance of the unified model is statistically significant in the terrorist events domain. These results ultimately serve to demonstrate the usefulness of event recognition for IE.

In addition to empirically demonstrating the usefulness of event recognition in IE, this chapter also presented analyses of features used by our models and discussed their strengths and weaknesses through an analysis of their errors. The feature analysis illustrated the need for semantic class information and local context in the form of lexico-syntactic patterns for certain types of event roles. Additionally, we found that separating the set of features used by the two components of our model was beneficial for some of the event roles. The analysis

Table 6.28: Summary of best performing models

Terrorist Events (MUC-4)															
	PerpInd			**PerpOrg**			**Target**			**Victim**			**Weapon**		
Model	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*	*P*	*R*	*F*
$\text{GLACIER}_{\text{SN}}$ 0.4	.51	.58	.54	.34	.45	.38	.42	.72	.53	.55	.58	.56	.57	.53	.55
AutoSlog-TS	.33	.49	.40	.52	.33	.41	.54	.59	.56	.48	.54	.51	.38	.45	.41
NB-Ex	.36	.34	.35	.35	.46	.40	.53	.49	.51	.50	.50	.50	1.00	.05	.10
Sem-Ex	.34	.46	.39	.30	.55	.39	.24	.78	.36	.18	.48	.26	.34	.69	.45

Disease Outbreaks (ProMed)						
	Disease			**Victim**		
Model	*P*	*R*	*F*	*P*	*R*	*F*
$\text{GLACIER}_{\text{NN}}$-Hum	.41	.61	.49	.38	.52	.44
AutoSlog-TS	.33	.60	.43	.36	.49	.41
NB-Ex	.34	.59	.43	.47	.39	.43
Sem-Ex	.12	.70	.20	.12	.65	.20

of examples where we supercede the baselines demonstrated that our unified probabilistic model can exploit event sentence information in almost 41% of the cases to overcome weak local evidence. Finally, an error analysis in Section 6.11 illustrated that improving the accuracy of the the localized text extraction component can have a large impact in further improving the overall performance of our IE models.

CHAPTER 7

CONCLUSIONS AND FUTURE DIRECTIONS

This chapter summarizes the conclusions and contributions of this research, and discusses possible future directions of this research. Section 7.1 presents the conclusions and observations presented in this research. Section 7.2 reiterates the scientific contributions of the dissertation. Finally, Section 7.3 discuses several avenues for future work that arise from various parts of this research.

7.1 Research Summary

The goal of this research has been to address the weaknesses of current IE systems that tend to rely on evidence of event role fillers solely in the local context surrounding candidate phrases in text. These systems look for direct evidence of role fillers, and are unable to deal with cases where the reader is expected to make inferences based on additional information present in the global context. To enable the correct extraction of role fillers in all of these cases we hypothesized that the IE system needs to use information from the wider context, in addition to information in the local context. This is the basic premise behind this dissertation.

To incorporate information from the wider sentential context, this dissertation takes an "event detection" approach. The idea is that if the IE model can identify event sentences, then this information could be used to strengthen the weaker local contextual clues. For example, if we can reliably detect that a sentence is describing a disease outbreak event, then the mention of an animal (e.g., *cow*) in this sentence can lead the system to reliably infer that that animal is the victim of that disease outbreak. A combination of such local clues and event information from the wider context can add up to reliable identification of event role fillers.

This research describes an approach incorporating two components to capture the two types of evidence in the local and wider contexts. One component's sole task is to identify sentences containing events of interest. This component uses a combination of features in

the wider context to determine if a sentence is discussing an event of interest. The second component then relies on local features surrounding a candidate phrase to decide if it should be extracted as an event role filler. This local component can capitalize on the information from the event sentence recognizer to make inferences about event role fillers, even in cases where the local evidence may be inconclusive in isolation.

The dissertation first presents a pipelined model that incorporates these two components as a cascaded system. In this pipelined approach, the output of the sentential event recognizer is fed as input to the localized text extraction system. The sentential event recognizer is implemented as a sentence classifier that uses features associated with each sentence to assign an event/nonevent label to it. The localized text extraction component is a pattern-based extraction module that learns extraction patterns based solely on the semantics of their extraction. This research shows that a combination of these two components in a pipeline results in an effective IE system.

The pipelined approach for combining the two components, however, has some drawbacks. Firstly, it is difficult for the localized text extraction component to extract event roles from any sentence that gets incorrectly discarded by the sentential event recognizer as a nonevent sentence. Thus, the accuracy of the sentential component can strongly affect IE performance, making any gains from its use washed out by losses from its faults. The main problem is that the two components independently make discrete decisions, and this all-or-nothing nature of the event sentence recognition prevents the extraction component from even trying to use strong local evidence. The second issue with this model is its use of extraction patterns as local contextual evidence. Many other types of clues can be useful as "weaker" evidence for making inferences about event roles. Additionally, various types of weaker clues can together be more discerning than a single complex clue. Thus, we need another model to overcome these drawbacks of the pipelined approach.

To overcome the drawbacks of the pipelined model, a unified probabilistic model for IE is introduced. This model has the same two components as the pipelined approach, but rather than making discrete decisions independently, the two components generate probabilities that are combined into a joint probability upon which its decisions are based. The sentential event recognizer computes the probability of a sentence being an event sentence, and the extraction component (called the plausible role-filler recognizer in this model) computes the probability of event role filler strings. The two probabilities a combined into a single joint probability of a sentence being an event sentence, and a phrase being an event role filler. The model then make extraction decisions about noun phrases using this joint probability.

The probabilities in this model are based on several different types of contextual features associated with the sentences and with the noun phrases. Machine learning classifiers are employed for estimating these probabilities.

An evaluation of these two models is performed to empirically demonstrate the usefulness of event information in the wider context. The models are applied to standard IE data sets for two types of events — terrorist events and disease outbreaks — and compared with baseline IE systems that rely solely on the local context for event role extraction. The evaluation shows that the unified probabilistic model GLACIER$_{\text{SVM/NB}}$ using an SVM for the sentential event recognizer and a Naïve Bayes classifier for the plausible role-filler recognizer achieves the best performance for the terrorist events. For the disease outbreaks events the GLACIER$_{\text{NB/NB}}$ using Naïve Bayes classifiers for both components and using additional sentence-level human annotations achieves the best performance. Additionally, the research also shows that a fully automated pipelined model using weakly supervised methods — a self-trained sentence classifier and a semantic affinity pattern learner — can match the performance of an IE system that requires the oversight of a human expert.

The take-home message from this work is that event-based IE can benefit from an explicit event recognition model or component. Going beyond the local evidence through sentential event recognition can enable an IE model to make inferences about event role fillers, which would not be possible with only the local evidence. Furthermore, the use of a unified probabilistic model to incorporate sentential event recognition is shown to work better than a pipelined approach.

7.2 Research Contributions

This research demonstrates that event recognition in the wider context can be beneficial for IE. We observe that humans can effortlessly identify role players of events in text documents even when there is a lack of direct evidence in the local context to that effect. Such inferences are usually based on preconceived notions about the event and its role players. For an IE system to identify event role players in cases lacking direct evidence, it must have this ability to make inferences based on a combination of weaker clues. This research hypothesizes that an automated system can make such leaps of inference by detecting event descriptions in regions of text, and combining this information with local contextual clues. By detecting an event description in a region of text, an IE model can make more confident decisions about event role fillers, even if they appear with weak local evidence.

This research shows that making joint decisions through a probabilistic model is preferable

over a segregated pipelined model. In the course of this research, we find that separating the two components into two independent modules that make discrete decisions about event sentences and event role extractions has some pitfalls. The event role extraction component is unable to leverage strong local evidence in cases where the sentential event recognizer discards sentences with weaker global evidence of an event description. A unified probabilistic model combining the two components is shown to overcome the shortcomings of the pipelined approach.

Several machine learning classifier-based approaches (supervised as well as weakly supervised) are shown to be suitable for identifying event sentences in text. This research investigates several approaches for recognizing event descriptions in text. Several training strategies are investigated covering various levels of supervision. For example, a self-training strategy using only relevant and irrelevant document and a set of seed extraction patterns is developed for classifying sentences as event or nonevent sentences. Other weakly supervised approaches, such as multiple instance learning, were also investigated. A comparative evaluation of these is presented in this work. We show that a weakly supervised IE model is effective on two IE tasks. It matches the performance of an existing pattern-based IE system that benefits from human expert oversight. Our best-performing unified probabilistic model is a fully supervised classifier-based model.

7.3 Future Directions

This research answers many questions regarding the effective use of event sentences in event-based IE. In addition, it also opens up several avenues for interesting research in the future. Some of these ideas are described here.

7.3.1 Exploring Region Spans

This research has used event recognition at the sentence-level for achieving better IE performance. Sentence boundaries were used to demarcate "regions" in this work, because they are well-defined and convenient to use. However, it is possible that larger region sizes could work better. For example, consider the following sentences:

There was an explosion in Taipei yesterday. Three people have died.

Note that the second sentence has no evidence indicating the type of event. Thus, the correct extraction of *three people* as the victim of the incident in Taipei would require us to peer across sentence boundaries and determine that the previous event sentence alters how we should perceive the sentence following it.

This argues for extending event recognition beyond sentence boundaries to achieve better event role filler extraction. However, this apparently simple modification raises a number of questions and presents several open challenges that can be explored in future research.

One challenge is in determining the an ideal region size for a task or event type. It would be interesting to determine if there is an optimal region size for achieving best performance in a particular domain. Perhaps, the optimal size of the region depends on the specific type of event or the specific type of document or report under consideration.

An interesting avenue for exploration is a dynamic region span, whose optimal size can be determined at runtime for given documents, event types or event roles, based on characteristics of the input text. This exploration can take into account the variations in the types of event descriptions, such as detailed events stories and fleeting references to relevant events. Additionally, the research can explore discourse analysis and the strength of discourse relations across sentences in determining region boundaries. Thus, enhancements in event recognition can be achieved with several directions of future research in determining region spans for event descriptions.

7.3.2 Discourse Information for IE

In addition to the several avenues of future directions of this research applied in new applications, we can also see research opportunities in enhancing our existing IE models. One such direction explores the use of discourse analysis. Sentential event recognizers developed in our IE models apply to each sentence independent of the other sentences. Our sentential components pretend that there is no connection between sentences in a document. However, there are always implicit discourse links between sentences in a document. By incorporating such discourse information into event sentence recognition, we can potentially improve its recognition accuracy. The idea is that if two sentences are linked through discourse (e.g., one sentence is an "elaboration" of the first), then an event sentence classification of one sentence can indicate a similar label for the linked sentence.

One way to incorporate discourse is to create a graph of the sentences in a document, with each node representing a sentence in the document, and each edge representing a discourse link, such as those identified by a discourse parser. Weights can be assigned to the edges based on the number and types of links connecting two sentences. Graph-based methods, such as PageRank or graph Mincuts, is then applied to the graph to identify event/nonevent sentences using the discourse information.

Additionally, the graph-based methods can be combined with our classification methods to enable the use of discourse links along with various contextual features for a better

sentential event recognizer. The probabilities from the classifier can be used to initialize the node weights in the graph-based algorithm. Alternatively, the discourse links could be added as features to the sentence classifier to help improve its performance.

7.3.3 Event Role Analysis

An analysis of the various event roles in an event presents another opportunity of enhancing our existing IE models. One of our findings in this research has been that different event roles "respond" differently to each new IE technique. For example, in Chapter 6 we saw that after applying an IE model to the test data some event roles showed improvement, while others were negatively affected. This difference in behavior across event roles stems from differences in the semantics of role fillers. The role fillers of the *Victim* event role in the terrorist events data are typically humans, and hence are easily confused with other human entities, who are not role fillers for this particular event role. An event role like *Weapon* on the other hand has role fillers that are rather unique weapon words and are not easily confused with other types of role fillers. Thus the techniques that work well for these two event roles can vary.

An interesting avenue for future work is to perform an analysis of the various types of event roles, in an attempt to characterize them and the IE techniques applied to extract them. Such an analysis can then aim to categorize the types of IE techniques or methods and then predict which of these work better for specific types of event roles. Such an analysis can help build an IE system with different techniques for different event roles based on their characteristics determined from the event role analysis. Additionally, this can give us a better understanding of the properties of event roles (and their role fillers) that affect our choice of strategy for role filler extraction.

7.3.4 Exploiting Broad Coverage Corpora

Many pattern-based IE systems rely on domain-specific corpora for learning extraction patterns for a given IE task. Such domain specific corpora are expensive to create, and are not readily available for many types of events. The use of broad-coverage text corpora could make such systems easily portable to new domains and more effective for IE tasks. Event recognition techniques introduced in this research open up several avenues of research for existing IE systems to exploit broad-coverage corpora.

Our previous work [81] has shown how documents collected from the Web can be used to expand the coverage of a pattern-based IE system, without the need for annotating large corpora. The system used seed patterns to locate relevant segments of text on the Web and

extraction patterns having a high correlation with these relevant text segments were used to augment an existing pattern based IE system. The research presented in this dissertation can benefit such approaches by presenting techniques for event recognition in text.

Using the research presented in this dissertation, event recognition in text can be used to locate relevant text segments on the Web or in broad-coverage corpora with better coverage than seed patterns. With the discovered relevant text segments, patterns can then be learned from these for augmenting the learning processes of existing IE systems.

In fact, these patterns could be added to the localized text extraction component within the PIPER model to improve its own coverage. Not only can new event-specific texts be useful for other IE systems, they can also benefit the PIPER model in our own research as well. Thus, in addition to its use in a two phase IE system, an event recognition component can be shown to be beneficial for pattern learning from broad-coverage corpora.

In addition to pattern learning, the event recognition techniques can further be exploited in constructing new domain-specific corpora as relevant subsets of broad-coverage corpora. Regions of text in a broad-coverage corpus ranked using strategies presented in this dissertation, can prove useful for expanding existing (limited) domain-specific resources for a given IE task. Thus, event recognition presents possibilities beyond the two-stage IE models introduced in this research.

7.3.5 Event Recognition Across Applications

Because event-based IE is centered around events, the event recognition component introduced in this research is a natural fit for IE research. However, its usefulness can be imagined in other NLP applications as well, opening up several possibilities for novel enhancements. For instance, event recognition can play an important role in automatic text summarization and question answering for event-specific analysis.

Most summarization systems create generic summaries of documents with little regard to the goals of the reader. Any given document can be summarized in multiple ways and summaries, by definition, must discard some unimportant information in the process. Which information to discard is usually decided in a domain-independent manner (except in some cases, such as query-focused summarization). However, in cases a reader may consider certain domain-specific or event-specific information to be of interest in the generated summaries. The event recognition techniques employed in this research can prove valuable in such cases by weighting event-specific regions higher than the remaining text, and thus enabling the information in those regions to have a greater chance of appearing in the summaries.

In a similar vein, our event recognition strategies can potentially benefit the field of question answering. Question Answering (QA) systems designed for a specific domain can use our strategies to focus the search for answers to segments of text relevant to the specific domain. As an alternative approach, our strategies can be used to generate a subset of the corpus to be searched by the domain-specific QA system, in effect, resulting in potential speed and accuracy improvements. Thus, the techniques developed in this research can be used to benefit NLP applications outside of IE.

APPENDIX A

EVENT SENTENCE ANNOTATION GUIDELINES

This appendix lists the event sentence annotation guidelines that were provided to the annotators.

A.1 Task Definition

Most descriptions of events in documents, such as news articles, contain sentences explicitly describing the event, interspersed with other supplementary information not directly connected to the event. Our goal is to design a system that can identify sentences pertaining to specific events of interest in given input text. For example, if our concern is for *disease outbreak* events, the system should label those sentences in the text that specifically describe a disease outbreak.

To enable the training and evaluation of such a system, we require "gold standard" data with the ground truth annotated by human experts. Human annotators must label those sentences that describe each event of interest mentioned in the text. This document provides annotator guidelines for labeling sentences as being relevant to the specific events. The type of event of interest is defined in advance, based on specific applications or other external constraints.

For this annotation task, the annotator is provided with the document containing the text to be annotated along with the type of event to be annotated. The task for the human annotators is to annotate each sentence in the document with respect to this event-type. An annotator must assess each sentence in the text, and if the sentence contains the description of an event of interest, it should be assigned an *event sentence* label. If the sentence is not describing an event of interest, it should be labeled as a *nonevent sentence*. To establish a precise definition of an event description, the annotators should identify the following five types of sentences: *event incident*, *event consequence*, *event precursor*, *incidental information* or *not relevant*. Descriptions of these five types of sentences follow later in these guidelines. A sentence of the type *event incident* should be labeled as an

event sentence, while all the remaining four types of sentences should be labeled as *nonevent sentences*.

A.2 Recognizing Event Descriptions

The following paragraphs explain the five types of sentences, and the guidelines for recognizing them:

1. **Event Incident**

   Single incident at a single instant: Most commonly, an event consists of a single specific incident that occurs at a specific instant in time. For example, a suicide-bombing event occurs at the time that the explosion takes place. Any sentence describing the scene or the happenings of that time would clearly be considered a part of this event. Annotators should identify such sentences as *event incident* sentences.

   Multiple incidents over a time period: Many events, on the other hand, do not occur at a specific instant in time, but take place over a longer period of time and comprise of a number of smaller incidents that make up the event as a whole. For example, a disease outbreak event could consist of a number of incidents or reports of a disease occurring in a region. A sentence such as

   A case of Ebola was detected in Mercer county this morning.

   describes one incident, which may be part of a larger disease outbreak event. This could possibly be followed by a sentence such as:

   By the afternoon, several other cases had been discovered in the same region.

   which is describing yet another "incident" part of the same disease outbreak. Both of these sentences are part of the event of interest and should be recognized as *event incidents*.

   In general, annotators should annotate all related happenings or incidents that occur within the time-frame of the event as *event incidents*. In case of a bombing event, for example, the time-frame would be the instant that the explosion occurs. For a disease outbreak, on the other hand, the time-frame would start with the first few reports of a disease in a given region and would extend over a period of time.

   Incidents just before and immediately following the event: Often, however, it is difficult to define exactly when an event begins and when it ends. Further, it can be

argued that incidents leading up to the event and the immediate consequences of the event could be considered as part of the event. We, therefore, extend the time-frame of events to include incidents that occur **immediately before** the actual event (i.e., leading up to the event), and those that occur **immediately after** the event (i.e., direct consequences, and descriptions of the scene following the event). For example,

> *The security guards at the checkpoint saw the approaching truck laden with C4 explosives.*

describes the scene just before a bombing event. This should be recognized as an *event incident*.

Event references: Many times information of interest about a relevant event is not an "incident" within the event, but is mentioned in regards to the event as a whole, with possibly some additional information presented that may be unrelated to the event. For e.g.,

> *The bombing, which killed three people, has caused great unrest in the region and caused a minor revolt among the predominantly working-class population.*

primarily describes the broad consequences of the bombing event and does not specifically describe an incident that is part of the event. However, this sentence does provide specifics ("*killed three people*") of the event, with a reference to the event as a whole ("*this bombing*"). Such sentences are to be considered *event incidents*. Specifics of the event include entities participating in the event, the time or location of the event, etc. For instance, a phrase such as "*yesterday's bombing*" indicates the time that the event took place, and the sentence containing this phrase should be labeled as *event incident*. However, just references to the event, such as *the bombing* or *it* with no additional information about the specifics of the event, do not constitute an *event incident*.

2. Event Consequence

Broader, long-term consequences: News stories frequently describe the broad consequences or the larger impact of an event. Such consequences are typically described as incidents or happenings taking place later in time and sometimes at a different location. For example,

The bombing brought the peace process in the region to a grinding halt.

describes a larger consequence of a bombing event. Such incidents are recognized as *event consequences*.

It is important to note the difference between direct consequences immediately following the event and broader long-term consequences occurring later in time, occurring at a broader scale or at a different location. The former should be considered *event incidents* as per the previous guidelines (in 1). The latter occurs later in time, outside the time-frame defined for *event incidents* and, as such, should be considered *event consequences*.

Importance of causality: It is also important for the annotator to make a judgment about **causality**. Event consequences must be causally related to the event of interest. An incident that would have taken place irrespective of the occurrence of the event of interest cannot be considered an *event consequence*.

3. Event Precursor

Just like *event consequences* described above, we also come across incidents that are "precursors" to an event. The event of interest is a broader, long-term consequence of such precursor incidents. Annotators must recognize sentences describing such incidents as *event precursors*. For example, a sentence such as:

Members of the terrorist group said that the bombing in Tel Aviv was in direct retaliation to the Israeli air-strikes in the Gaza Strip.

describes the bombing event as being causally related to air-strikes occurring earlier in time. Here the bombing event is the event of interest and the air-strikes are, therefore, considered "precursor" incidents.

Annotators should clearly distinguish *event precursors* from *event incidents* that occur immediately before the relevant event. As specified earlier in the guidelines (in 1), incidents occurring immediately before the relevant event are considered to be part of the event, and are labeled as *event incidents*. Therefore, for an incident to be labeled an *event precursor*, the incident should occur outside the time-frame of the specific event.

Further, causality is an important factor to be considered in recognizing *event precursor* sentences. The event of interest must be causally related to all event precursors.

In other words, if the event of interest would have taken place irrespective of the occurrence of a preceding incident, then the preceding incident cannot be an *event precursor*.

4. Incidental Information

One of our observations from analyzing event descriptions was that, frequently to capture reader interest, writers include intriguing facts, titbits or properties of the location, persons or objects involved in the event. Sometimes, interesting facts about the date or time at which the event took place are described in the document. Usually, these sentences do not pertain specifically to the event or to the consequences of the event, but are details about the persons, objects, dates or locations playing a part in the event. For instance, consider

> *The explosion blew out all the windows of an empty building. The building was constructed in 1901, and had been scheduled for demolition next month.*

The first sentence clearly describes an *event incident*. The second sentence then presents some interesting information pertaining to the building mentioned in the first sentence. The second sentence is a factual statement and does not specifically describe an incident or happening. Factual sentences, such as these, about entities mentioned in the event should be recognized as *incidental information*.

Sometimes, the document contains additional information, not about a specific entity, but about the event itself. Such additional information is usually factual "meta-information" about the event, and does not describe an actual incident occurring as part of the event. In such cases, the sentence should be recognized as *incidental information*. For example, suppose the two sentences in the example above were followed by this sentence:

> *This bombing is similar to last year's explosion in the Sarojini Nagar Market.*

This would be considered *incidental information*, because it only provides additional factual "meta-information" about the bombing event described earlier. It is important to distinguish these cases from the "event references" cases described for the *event incident* label. These cases provide no specifics about the event of interest.

Thus, based on the above guidelines, annotators are instructed to ensure that any sentence recognized as *incidental information* meet at least one of the following two requirements:

(a) it refers to a specific entity in an *event incident* sentence, or

(b) it provides "meta-information" about a relevant event or incident described in the *event incident* sentence.

5. Not Relevant

Any sentence that does not pertain to a relevant event (and hence is not recognized as any one of the four previous sentence types) should be considered a *not relevant* sentence.

With these guidelines for the sentence types, annotators can label sentences that describe various aspects of an event of interest. Each *event incident* sentence should be assigned an *event sentence* label. Likewise all other sentence types should be labeled as *nonevent sentence*.

Additionally, annotators are advised to read the entire story completely, at least once (preferably twice), before assigning labels to sentences in the story. This requirement will enable the annotators to gauge the level of detail presented for each event in the story, and will make it easier to tease apart the differences in the sentence types.

A.3 Annotating Separate Events

We observe that a single document may contain a single event of interest, or may contain multiple events of interest. For each sentence that is annotated with *event sentence* label, the annotators are further instructed to attach an event number such as *event #1*, *event #2*, etc. indicating the specific event that the sentence belongs to. A set of sentences together describing a single event should all have the same event number.

Deciding the boundaries between different events within the same document is a potential issue with annotating separate event instances. In most cases, the descriptions of different events should be quite clear. However, there could be cases such as multiple bombings in the same city, or the outbreak of the same disease in multiple locations, which are not as clear-cut. Event numbers for these are left to the judgment of the annotator. The only guiding policy is to try to infer the intent of the writer in these cases. If the intent of the writer is to portray the incidents as separate events, then they should be annotated

as such. Conversely, if the intent of the writer is to portray the incidents as part of a single event, then they should get the same event number.

We acknowledge that annotating separate event is indeed a difficult task. Thus, to make the annotation task easier, a list of relevant events may be provided, in advance, along with each of the documents. In this case, the annotators are instructed to select one or more of these events from the list for each sentence labeled with the *event sentence* label.

A.4 Notes on Specific Events

For this annotation task, we will be focusing on two domains: *terrorism* and *disease outbreaks*. The task of the annotator is to annotate regions of text with respect to events in these two domains. Specific notes about these two domains are provided in this appendix.

A.4.1 Terrorist Events

The data to be annotated in this domain was selected from the MUC-3 and MUC-4 corpus [113, 114]. As a result, the definition of a relevant terrorist event follows that of the MUC-3 and MUC-4 IE tasks. Quoting from the MUC-3 proceedings:

> Relevant incidents are, in general, violent acts perpetrated with political aims and a motive of intimidation. These are acts of terrorism.

A relevant event can be one of six different types: *attack*, *bombing*, *kidnapping*, *arson*, *robbery*, and *forced work stoppage*. Additionally, these can be at different stages of execution: *threatened*, *attempted* and *accomplished*. Thus, in this framework, even threats of aggression made against an individual or organization are considered terrorist events, and should be annotated.

A.4.2 Disease Outbreak Events

For disease outbreaks domain, the documents to be annotated were chosen from the data set created by Sean Igo for use in recent Information Extraction tasks [86, 82]. The documents were obtained from *Promed mail*,[1] an online reporting system for outbreaks of emerging infectious diseases. A relevant event in this domain consists of the outbreak of a specific disease in a region or country. Quoting from the annotation guidelines for the IE task:

---

[1] http://www.promedmail.org

> What is an "outbreak?" A template should be created only for reports of a specific outbreak. General descriptions of outbreaks (e.g., "a widespread outbreak of anthrax would cripple society") are NOT relevant.
>
> Only current/recent outbreaks are of interest, especially those ongoing at the time the article was written. Any outbreak described as having happened more than a year before the article is written should be avoided. There is a bit of leeway — for instance, if an article was written in July and refers to "last summer," that is close enough, and similarly 367 days ago is close enough. Use your judgment on these borderline cases.
>
> Outbreaks do not need to have identifiable victims. Outbreaks do need implicit or explicit victims, or have a confirmed organism or disease that creates a potential health hazard. The latter would include terrorist actions, such as anthrax mailings, even if no one was infected.
>
> Multiple outbreaks of the same disease in the same country, if authorities claim they are unrelated, should be marked up as separate outbreaks.

Thus, in summary, only recent outbreaks are of interest to us. Further, outbreaks may not have identifiable victims or diseases. Lastly, if multiple outbreaks of a disease occurring in a country or region are considered unrelated by authorities, then they constitute separate outbreak events.

A.5 Addendum

After concluding several training sessions with the annotators, the key points discussed in the guidelines and during the training session were summarized in a handout for the annotators. A copy of this handout is presented in Figure A.1.

Notes on Event Annotations

1. Event Sentence

Sentences included in this category definition contain the description of an event of interest, and specific details of that event. These are primarily controlled by an event timeline established by the annotators. Sentences in this category must meet at least one of the following criteria:

(a) **Establish event timeline. All incidents within this timeline AND relevant to the main event are labeled as event sentence.**

In the terrorism domain, the timeline typically starts with incidents leading up to the event (bombing, kidnapping, murder, etc.), followed by descriptions of the event and end with any related happenings immediately following the event.

In the disease outbreaks domain, the timeline typically starts with discovery of symptoms of a disease, followed by a diagnosis and ending with treatment/recovery/fatality.

(b) **A reference to the event ("it", "the bombing", etc.) AND the mention of an event specific.**

For each event, specific information of interest includes (but not limited to) *the date, location, physical objects, humans, animate objects (animals, plants, etc.), conditions, attributes, circumstances or actions involved in the event.*

In addition, speculation about possible victims or diseases (or other event specifics) is acceptable for an event sentence ("*the bombings were most likely perpetrated by the Al Qaeda or the FMLN*").

However, avoid inferring specific references to perpetrators, victims, targets or weapons from more general references ("*only one facility in the UK is capable of handling **such** animals*").

2. Non-Event Sentence

Sentences that do not meet the conditions of an *Event Sentence* should be labeled as *Non-Event Sentence*. These include sentences that meet the following criteria (but not limited to):

(a) **Event Precursor. Events/incidents that take place earlier in time AND there is a causal relationship (implied in the document) between the earlier event and the current event.**

(b) **Event Consequence. Events/incidents that take place later in time AND the document implies that they are causally related to the current event.**

(c) **Factual information about an entity mentioned in the event (one of the event specifics).**

Must contain a reference to one of the entities (one of the event specifics) mentioned in an event incident, and be some additional information about that entity. Note that this is different from 1(b), in that this does not contain a reference to the relevant event.

(d) **Speculation about possible future occurrences and about things that "could have been".**

Figure A.1: Handout summarizing the annotation guidelines

APPENDIX B

SEMANTIC CLASS MAPPING FOR SEMANTIC AFFINITY

The semantic affinity measure used in the PIPER model for IE requires a manually created mapping between semantic classes and event roles. This appendix presents the mapping between semantic classes and event roles (SEMCLASS → EvRole) that was used for all our experiments involving the semantic affinity metric in the two domains. For terrorist events, the following mapping was used for the five event roles evaluated in the IE experiments:

AERIAL-BOMB → Weapon
CUTTING-DEVICE → Weapon
EXPOSIVE → Weapon
GUN → Weapon
PROJECTILE → Weapon
STONE → Weapon
WEAPON → Weapon
UNSPECIFIED-WEAPON → Weapon
OTHER-WEAPON → Weapon

STRUCTURE → Target
UNSPECIFIED-STRUCTURE → Target
BUILDING → Target
UNSPECIFIED-BUILDING → Target
CHURCH → Target
CIVILIAN-RESIDENCE → Target
COMMERCIAL-BUILDING → Target
FINANCIAL-INSTITUTION → Target
OTHER-BUILDING → Target
UTILITY-STRUCTURE → Target
SITE → Target
TRANSPORTATION-ROUTE → Target
OTHER-STRUCTURE → Target
VEHICLE → Target
MEDIA-ORGANIZATION → Target

FAMILY-RELATION → Victim
NATIONALITY → Victim
CIVILIAN → Victim
UNSPECIFIED-CIVILIAN → Victim
DIPLOMAT → Victim
POLITICIAN → Victim
GOVT-OFFICIAL → Victim
LAW-ENFORCEMENT → Victim
LEGAL-OR-JUDICIAL → Victim
SCIENTIST → Victim
OTHER-CIVILIAN → Victim

VIOLENT-CRIMINAL → PerpInd

TERRORIST-ORGANIZATION → PerpOrg

For the disease outbreaks events, the following mapping was used:

Animate → Victim	Other-Human → Victim
Unspecified-Animate → Victim	Animal → Victim
Human → Victim	Unspecified-Animal → Victim
Unspecified-Human → Victim	Amphibian → Victim
Family-Relation → Victim	Bird → Victim
Nationality → Victim	Fish → Victim
Occupation → Victim	Mammal → Victim
Civilian → Victim	Mollusk → Victim
Unspecified-Civilian → Victim	Reptile → Victim
Diplomat → Victim	Other-Animal → Victim
Politician → Victim	Plant → Victim
Govt-Official → Victim	Other-Animate → Victim
Law-Enforcement → Victim	Number → Victim
Legal-Or-Judicial → Victim	
Media-Worker → Victim	Microorganism → Disease
Scientist → Victim	Unspecified-Microorganism → Disease
Other-Civilian → Victim	Bacterium → Disease
Military → Victim	Archaeon → Disease
Criminal → Victim	Rickchlam → Disease
Unspecified-Criminal → Victim	Neuroamine → Disease
Violent-Criminal → Victim	Fungus → Disease
Nonviolent-Criminal → Victim	Virus → Disease
Person-Name → Victim	Other-Microorganism → Disease
Male-Name → Victim	Bioact → Disease
Female-Name → Victim	Poison → Disease
Neutral-Name → Victim	Acqabn → Disease
Title → Victim	Disease → Disease
Job-Title → Victim	Unspecified-Disease → Disease
Academic-Title → Victim	Specific-Disease → Disease
Other-Title → Victim	Other-Disease → Disease

The left side of the mapping (i.e., on the left side of →) lists a semantic class from the semantic dictionary used by the Sundance system. Likewise, the right side of the mapping (i.e., on the right side of →) lists an event role for the given event type. Any of the semantic classes from Sundance not listed in this mapping get automatically mapped to a special "Other" event role.

APPENDIX C

OVERVIEW OF EXTRACTION PATTERNS

All of the extraction patterns used in this research are based on the AutoSlog-TS system [92]. These patterns are lexico-syntactic patterns that rely on a shallow parse of text. This appendix describes these patterns in more detail.

Information extraction (IE) patterns are lexico-syntactic patterns that represent expressions identifying role relationships in text. For example, a pattern of the form "<subj> ActVP(*damaged*)" extracts the subject of active voice instances of the verb "damaged" as an entity (NP) that caused damage. Similarly, the pattern "<subj> PassVP(*damaged*)" extracts the subject of passive voice instances of "damaged" as the object that was damaged. Thus, "<subj> ActVp" and "<subj> PassVp" represent two types of patterns that represent roles of NPs mentioned in text.

AutoSlog-TS defines seventeen different types of such patterns, which are shown in Table C.1. An instantiated pattern of each type, and a text snippet matching the pattern are also shown in the table. In these patterns, PassVP refers to passive voice verb phrases (VPs), ActVP refers to active voice VPs, InfVP refers to infinitive VPs, and AuxVP refers to VPs where the main verb is a form of "to be" or "to have." Subjects (subj), direct objects (dobj), PP objects (np), and possessives can be extracted by the patterns.

All seventeen patterns are used for terrorist event IE. However, for the disease outbreaks IE task a slightly different set of pattern types is used. In this domain, the possessive pattern type ("<possessive> NP") is not used; and four additional pattern types (shown at the bottom of Table C.1) are used in the IE systems. The "<ordinal> NP" pattern type represents patterns containing an ordinal (e.g., *first case*, *third patient*, etc.), while the "<number> NP" pattern type represents patterns containing a numeric modifier (e.g., *one case*, *three patients*, etc.).

Table C.1: AutoSlog-TS pattern types and sample IE patterns

Pattern Type	Example Pattern	Example Text
<subj> PassVP	<victim> were murdered	*Two school teachers were brutally murdered yesterday...*
<subj> ActVP	<perp> killed	*Armed gunmen killed three innocent bystanders...*
<subj> ActVP Dobj	<weapon> caused damage	*A bomb caused massive damage to buildings in the...*
<subj> ActInfVP	<perp> tried to kill	*An assailant tried to kill the attorney on his way to...*
<subj> PassInfVP	<weapon> was aimed to hit	*The missile was aimed to hit the home of a prominent...*
<subj> AuxVP Dobj	<victim> was target	*The chairman was the target of an attack by unidentified...*
<subj> AuxVP Adj	<perp> were responsible	*FMLN operatives were responsible for several attacks...*
ActVP <dobj>	bombed <target>	*...bombed the Embassy yesterday causing widespread panic.*
InfVP <dobj>	to attack <target>	*...made arrangements to attack the convoy just as it...*
ActInfVP <dobj>	planned to bomb <target>	*...insurgents planned to bomb a military base in Kabul.*
PassInfVP <dobj>	was designed to kill <victim>	*The IED was designed to kill Iraqi soldiers...*
Subj AuxVP <dobj>	weapon was <weapon>	*The weapon was an AK-47 rifle with armor-piercing...*
NP Prep <np>	attack against <target>	*It was the third attack against US soldiers since...*
ActVP Prep <np>	attacked with <weapon>	*The militants attacked the village with explosives...*
PassVP Prep <np>	were killed with <weapon>	*The men were killed with bullets from a 9mm revolver...*
InfVP Prep <np>	to attack with <weapon>	*...are making plans to attack the convoy with explosives.*
<possessive> NP	<victim>'s murder	*The judge's murder has stunned the people...*
Additional pattern types, used for Disease Outbreak events		
<subj> ActVp Adjp	<victim> tested positive	*The three patients tested positive for Swine Flu...*
Subj PassVp Prep <np>	disease found in <victim>	*...taken to control the disease found in the three patients.*
<ordinal> NP	<ordinal victim>	*This is the third case of Ebola observed within the past week...*
<number> NP	<number victims>	*About 75 H1N1 cases have been reported reported in Salt Lake...*

APPENDIX D

LIST OF EXTRACTIONS WITH WEAK LOCAL EVIDENCE

An analysis of GLACIER in Section 6.10 (Chapter 6) showed that in many cases our model is able to exploit the event sentence information from the wider context to correctly extract event role fillers even in cases that lack strong local evidence. This appendix lists these specific cases from the test data that were found in our analysis. In each of the following sentences, $\text{GLACIER}_{\text{SVM/NB}}$ correctly extracts an event role filler despite the lack of direct evidence in the local context of the extracted text.

1. THE MNR REPORTED ON 12 JANUARY THAT HEAVILY ARMED MEN IN CIVILIAN CLOTHES HAD INTERCEPTED A VEHICLE WITH OQUELI AND FLORES ENROUTE FOR LA AURORA AIRPORT AND THAT THE TWO POLITICAL LEADERS HAD BEEN KIDNAPPED AND WERE REPORTED MISSING.

   PerpInd = "HEAVILY ARMED MEN"

2. THE BODIES OF HECTOR OQUELI, UNDERSECRETARY OF THE NATIONAL REVOLUTIONARY MOVEMENT [MNR] OF EL SALVADOR, AND GILDA FLORES, A MEMBER OF GUATEMALA'S SOCIAL DEMOCRATIC PARTY, WERE FOUND IN CUILAPA, GUATEMALA, NEAR THE BORDER WITH EL SALVADOR, THE RELATIVES OF ONE OF THE VICTIMS HAVE REPORTED.

   Victim = "GILDA FLORES"

3. SINCE THE MEETING, IT HAS INCREASED ITS DETENTIONS AND ATTACKED THE RESIDENCES OF LEADERS OF THE SOCIAL CHRISTIAN PEOPLE'S MOVEMENT AND THE UDN [NATIONALIST DEMOCRATIC UNION], UNIVERSITY, FECMASAN [MSGR OSCAR ARNULFO ROMERO FEDERATION OF COMMITTEES OF MOTHERS AND RELATIVES], AND FENASTRAS.

   Target = "UNIVERSITY"

4. TWO DAYS LATER, ON 16 AUGUST, DUE TO DEATH THREATS THE EIGHTH PUBLIC ORDER JUDGE IN MEDELLIN, WHO WAS INVESTIGATING THE GOVERNOR OF ANTIOQUIA'S MURDER, RESIGNED.

   Victim = "THE EIGHTH PUBLIC ORDER JUDGE"

5. I AM SPEAKING ON BEHALF OF THE SALVADORAN ARMED FORCES THAT IS ON ITS FEET AT 0115 ON 12 NOVEMBER, CONFRONTING A TREACHEROUS ATTACK BY THE FMLN [FARABUNDO MARTI NATIONAL LIBERATION FRONT] TERRORISTS WHO AT 2000 ON 11 NOVEMBER – DISPLAYING THEIR WARMONGERING STRATEGY AND WITHOUT CARING ABOUT THE SUFFERING OF THE SALVADORAN PEOPLE – CARRIED OUT TREACHEROUS ACTIONS IN WHICH THE CIVILIAN POPULATION WERE THE ONLY ONES WHO HAVE SUFFERED.

   PerpInd = "TERRORISTS"

6. IN HOMAGE TO A TRADE UNION LEADER KILLED 13 DAYS AGO ALONG WITH NINE OTHER WORKERS IN A DYNAMITE EXPLOSION IN THE CAPITAL CITY ATTRIBUTED TO THE ULTRARIGHTIST DEATH SQUADS.

   Victim = "NINE OTHER WORKERS"

7. THE OFFENSIVE LAUNCHED BY THE FMLN ON 11 NOVEMBER, THE MOST IMPORTANT ONE SINCE 1981, HAS ALREADY RESULTED IN 139 DEAD AND 313 WOUNDED, INCLUDING SOLDIERS, REBELS, AND CIVILIANS, GOVERNMENT SOURCES REPORTED TODAY.

   Victim = "CIVILIANS"

8. ACCORDING TO REUTER, ATTEMPTS WERE MADE TO STORM PRESIDENT ALFREDO CRISTIANI'S OFFICIAL AND PERSONAL RESIDENCES;

   Target = "PERSONAL RESIDENCES"

9. AT 0430 (1030 GMT) TODAY, THE FMLN GUERRILLAS INITIATED HEAVY FIGHTING AROUND THE SHERATON HOTEL, WHERE OAS SECRETARY GENERAL JOAO BAENA SOARES WAS STAYING.

   Victim = "OAS SECRETARY GENERAL JOAO BAENA SOARES"

10. SEVERAL REBELS HAVE SAID THAT FACUNDO GUARDADO ("COMMANDER ESTEBAN"), ONE OF THE FOUNDERS OF THE FARABUNDO MARTI NATIONAL LIBERATION FRONT AND ONE OF THE LEADERS OF THE OFFENSIVE THAT BEGAN IN THE CAPITAL ON 11 NOVEMBER, IS SOMEWHERE NEAR THE HOTEL.

    PerpOrg = "THE FARABUNDO MARTI NATIONAL LIBERATION FRONT"

11. AS YOU WILL REMEMBER, THE GOVERNMENT HAS MET WITH MUCH INTERNATIONAL DISREPUTE AND ISOLATION BECAUSE OF THE BRUTAL ASSASSINATION OF JESUIT PRIESTS AND THE BOMBING OF THE CIVILIAN POPULATION.

    PerpOrg = "THE GOVERNMENT"

12. AFTER REGISTERING, BERNARDO JARAMILLO ATTENDED THE FUNERAL OF FELLOW PARTY MEMBER, VLADIMIR ESCOBAR, A SOACHA MUNICIPAL OFFICIAL WHO WAS SHOT BETWEEN THE EYES AND KILLED THIS WEEK.

    Victim = "VLADIMIR ESCOBAR"

13. THE STATE OF SIEGE, THE ASSASSINATION OF A SOCIAL DEMOCRATIC LEADER HECTOR OQUELI, THE DOUBTFUL AND INSUFFICIENT INITIAL RESULTS IN THE CASE OF THE JESUIT PRIESTS, THE ABSENCE OF CONDITIONS FOR THE POLITICAL ACTIVITY OF THE OPPOSITION, THE CONTINUATION OF THE PERSECUTION OF CHURCHES, AND THE LACK OF FREEDOM OF SPEECH THROUGH THE DIRECT OR COVERT GAGGING OF THE PRESS SHOW THAT CRISTIANI'S GOVERNMENT IS CARRYING OUT A POLICY OF CONFRONTATION WITH ALL THE COUNTRY'S SOCIAL AND POLITICAL FORCES.

    PerpOrg = "CRISTIANI'S GOVERNMENT"

14. THE ARMED FORCES PRESS COMMITTEE (COPREFA) REPORTED THAT ON THE MORNING OF 1 JUNE GUERRILLAS, USING MORTARS AND RIFLES, ATTACKED A CEL SUB-STATION IN THE NATIVIDAD CANTON IN THE WESTERN DEPARTMENT OF SANTA ANA.

    Weapon = "RIFLES"

15. THE ARMED FORCES PRESS COMMITTEE (COPREFA) REPORTED THAT ON THE MORNING OF 1 JUNE GUERRILLAS, USING MORTARS AND RIFLES, ATTACKED A CEL SUB-STATION IN THE NATIVIDAD CANTON IN THE WESTERN DEPARTMENT OF SANTA ANA.

    Weapon = "MORTARS"

16. ACCORDING TO MILITARY AUTHORITIES, A LIEUTENANT AND FIVE CIVILIANS WHO WERE PASSING THROUGH THE AREA AT THE TIME WERE INJURED BY THE EXPLOSION.

    Victim = "FIVE CIVILIANS"

17. ACCORDING TO THE MILITARY REPORTS, CORPORAL ALEXANDER MOLINA GRANADOS WAS KILLED AS HE WAS TRYING TO DEFUSE A MINE PLACED BY THE INSURGENTS ON LA NORIA BRIDGE, IN SAN MARCOS LEMPA.

    Target = "LA NORIA BRIDGE"

18. NEAR THE END OF THE CAST, THE ANNOUNCER ADDS TO THE INITIAL REPORT ON THE EL TOMATE ATTACK WITH A 3-MINUTE UPDATE THAT ADDS "2 INJURED, 21 HOUSES DESTROYED, AND 1 BUS BURNED."

    Target = "1 BUS"

19. NEAR THE END OF THE CAST, THE ANNOUNCER ADDS TO THE INITIAL REPORT ON THE EL TOMATE ATTACK WITH A 3-MINUTE UPDATE THAT ADDS "2 INJURED, 21 HOUSES DESTROYED, AND 1 BUS BURNED."

    Target = "21 HOUSES"

20. THESE TERRORIST ATTACKS TOOK PLACE 1 DAY AFTER THE SERIOUS ATTACK LAUNCHED AT THE 2D ARMY DIVISION HEADQUARTERS IN BUCARAMANGA, WHICH RESULTED IN SEVEN PEOPLE INJURED AND CONSIDERABLE PROPERTY DAMAGE, AFFECTING NINE HOMES.

    Target = "NINE HOMES"

21. GUERRILLAS OF THE FARC AND THE POPULAR LIBERATION ARMY (EPL) ATTACKED FOUR TOWNS IN NORTHERN COLOMBIA, LEAVING 17 GUERRILLAS AND 2 SOLDIERS DEAD AND 3 BRIDGES PARTIALLY DESTROYED.

    PerpOrg = "THE FARC"

22. EPL [POPULAR LIBERATION ARMY] GUERRILLAS BLEW UP A BRIDGE AS A PUBLIC BUS, IN WHICH SEVERAL POLICEMEN WERE TRAVELING, WAS CROSSING IT.

    Victim = "SEVERAL POLICEMEN"

23. EPL [POPULAR LIBERATION ARMY] GUERRILLAS BLEW UP A BRIDGE AS A PUBLIC BUS, IN WHICH SEVERAL POLICEMEN WERE TRAVELING, WAS CROSSING IT.

    Target = "A PUBLIC BUS"

24. MEMBERS OF THE BOMB SQUAD HAVE DEACTIVATED A POWERFUL BOMB PLANTED AT THE ANDRES AVELINO CACERES PARK, WHERE PRESIDENT ALAN GARCIA WAS DUE TO PARTICIPATE IN THE COMMEMORATION OF THE BATTLE OF TARAPACA.

    Victim = "PRESIDENT ALAN GARCIA"

25. THEY ALSO REPORTED THAT THE ANTEL [NATIONAL ADMINISTRATION FOR TELECOMMUNICATIONS] OFFICE, COURT OFFICES, AND COMMUNITY CENTER OF DULCE NOMBRE DE MARIA WERE DYNAMITED BY THE FMLN EARLY THIS MORNING.

    Target = "OFFICE"

26. THEY ALSO REPORTED THAT THE ANTEL [NATIONAL ADMINISTRATION FOR TELECOMMUNICATIONS] OFFICE, COURT OFFICES, AND COMMUNITY CENTER OF DULCE NOMBRE DE MARIA WERE DYNAMITED BY THE FMLN EARLY THIS MORNING.

    Target = "COURT OFFICES"

27. THEY WERE ASSASSINATED ALONG WITH 12 VENEZUELAN CITIZENS WHEN THE VENEZUELAN ARMY MISTOOK THEM FOR NATIONAL LIBERATION ARMY REBELS IN CANO COLORADA, ARAUCA DEPARTMENT, ALONG THE VENEZUELAN BORDER.

    PerpOrg = "THE VENEZUELAN ARMY"

28. IN ADDITION, 4 TRAINS BELONGING TO THE SALVADORAN NATIONAL RAILROAD, FENADESAL, WERE BLOWN UP, AND 10 VEHICLES AND 5 BUSES WERE PARTIALLY DAMAGED.

    Target = "10 VEHICLES"

29. REGARDING DAMAGE TO THE ELECTRICITY SYSTEM, THE FMLN KNOCKED DOWN 46 TOWERS AND 136 HIGH-TENSION POWER POLES AND DAMAGED 11 TRANSFORMERS.

    Target = "136 HIGH-TENSION POWER POLES"

30. IN THE WAR AGAINST PUBLIC PROPERTY – THE PILLAR OF A WAR ECONOMY – THE MILITARY SOURCE REPORTED THAT SIX MAYORIAL OFFICES AND SIX NATIONAL ADMINISTRATION FOR TELECOMMUNICATIONS OFFICES WERE DAMAGED IN VARIOUS TOWNS AND SIX JUNCTION BOXES WERE SABOTAGED.

    Target = "SIX MAYORIAL OFFICES"

31. THE SAME SOURCES REPORTED THAT THOSE INJURED, AN OFFICER AND TWO NONCOMMISSIONED OFFICERS, WERE CROSSING THE BRIDGE IN A PRIVATE CAR WHEN A BOMB EXPLODED DESTROYING THE BRIDGE.

    Target = "A PRIVATE CAR"

32. THE SAME SOURCES REPORTED THAT THOSE INJURED, AN OFFICER AND TWO NONCOMMISSIONED OFFICERS, WERE CROSSING THE BRIDGE IN A PRIVATE CAR WHEN A BOMB EXPLODED DESTROYING THE BRIDGE.

Victim = "TWO NONCOMMISSIONED OFFICERS"

REFERENCES

[1] Agichtein, E., and Gravano, L. Snowball: Extracting Relations from Large Plain-Text Collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries* (San Antonio, TX, June 2000), pp. 85–94.

[2] Altomari, P., and Currier, P. Focus of TIPSTER Phases I and II. In *TIPSTER Text Program Phase II: Proceedings of the Workshop* (Vienna, VA, May 1996), pp. 9–11.

[3] Andrews, S., Tsochantaridis, I., and Hofmann, T. Support Vector Machines for Multiple-Instance Learning. In *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, Cambridge, MA, 2003, pp. 561–568.

[4] Appelt, D., Hobbs, J., Bear, J., Israel, D., Kameyama, M., and Tyson, M. FASTUS: A Finite-State Processor for Information Extraction from Real-World Text. In *Proceedings of the 13th International Joint Conference on Artifical Intelligence* (Chambéry, France, August 1993), pp. 1172–1178.

[5] Azzam, S., Humphreys, K., and Gaizaukas, R. Using Coreference Chains for Text Summarization. In *Proceedings of the Workshop on Coreference and Its Applications* (College Park, MD, June 1999), pp. 77–84.

[6] Barzilay, R., and Elhadad, M. Using Lexical Chains for Text Summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization* (Madrid, Spain, August 1997), pp. 10–17.

[7] Bates, D., Evans, R., Murff, H., Stetson, P., Pizziferri, L., and Hripcsak, G. Detecting Adverse Events Using Information Technology. *Journal of the American Medical Informatics Association 10*, 2 (March 2003), 115–128.

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

[9] Bunescu, R., and Mooney, R. Subsequence Kernels for Relation Extraction. In *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. MIT Press, Cambridge, MA, 2006, pp. 171–178.

[10] Bunescu, R., and Mooney, R. Learning to Extract Relations from the Web Using Minimal Supervision. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics* (Prague, Czech Republic, June 2007), pp. 576–583.

[11] Bunescu, R., and Mooney, R. Multiple Instance Learning for Sparse Positive Bags. In *Proceedings of the 24th International Conference on Machine Learning* (Corvallis, OR, June 2007), pp. 105–112.

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

[13] Callan, J. Passage-Level Evidence in Document Retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Dublin, Ireland, July 1994), pp. 302–310.

[14] Chieu, H., and Ng, H. Named Entity Recognition: a Maximum Entropy Approach using Global Information. In *Proceedings of the 19th International Conference on Computational Linguistics* (Taipei, Taiwan, August 2002), pp. 1–7.

[15] Chieu, H., Ng, H., and Lee, Y. Closing the Gap: Learning-Based Information Extraction Rivaling Knowledge-Engineering Methods. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (Sapporo, Japan, July 2003), pp. 216–223.

[16] Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S. Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (Vancouver, Canada, October 2005), pp. 355–362.

[17] Ciravegna, F. Adaptive Information Extraction from Text by Rule Induction and Generalisation. In *Proceedings of Seventeenth International Joint Conference on Artificial Intelligence* (Seattle, WA, August 2001), pp. 1251–1256.

[18] Ciravegna, F., and Lavelli, A. LearningPinocchio: Adaptive Information Extraction for Real World Applications. *Natural Language Engineering 10*, 2 (June 2004), 145–165.

[19] Clarke, C., Cormack, G., Kisman, D., and Lyman, T. Question Answering by Passage Selection (MultiText Experiments for TREC-9). In *Proceedings of the Ninth Text REtrieval Conference (TREC-9)* (Gaithersburg, MD, November 2000), pp. 673–683.

[20] Clarke, C., Cormack, G., and Lyman, T. Exploiting Redundancy in Question Answering. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New Orleans, LA, September 2001), pp. 358–365.

[21] Cohen, J. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement 20*, 1 (1960), 37–46.

[22] Collins, M. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, 1999.

[23] Collins, M., and Singer, Y. Unsupervised Models for Named Entity Classification. In *Proceedings of Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora* (College Park, MD, June 1999), pp. 100–110.

[24] Cowie, J. Automatic Analysis of Descriptive Texts. In *Proceedings of the First Conference on Applied Natural Language Processing* (Santa Monica, CA, February 1983), pp. 117–123.

[25] Cui, H., Sun, R., Li, K., Kan, M., and Chua, T. Question Answering Passage Retrieval Using Dependency Relations. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Salvador, Brazil, August 2005), pp. 400–407.

[26] Cullingford, R. *Script Application: Computer Understanding of Newspaper Stories*. PhD thesis, Yale University, January 1978.

[27] Culotta, A., and Sorensen, J. Dependency Tree Kernel for Relation Extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (Barcelona, Spain, July 2004), pp. 423–429.

[28] Cunningham, H., Humphreys, K., Gaizauskas, R., and Wilks, Y. TIPSTER-Compatible Projects at Sheffield. In *TIPSTER Text Program Phase II: Proceedings of the Workshop* (Vienna, VA, May 1996), pp. 121–123.

[29] Daumé, H., and Marcu, D. Bayesian Query-Focused Summarization. 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. 305–312.

[30] DeJong, G. Prediction and Substantiation: A New Approach to Natural Language Processing. *Cognitive Science: A Multidisciplinary Journal 3*, 3 (July 1979), 251–271.

[31] DeJong, G. An Overview of the FRUMP System. In *Strategies for Natural Language Processing*, W. Lehnert and M. Ringle, Eds. Erlbaum, Hillsdale, NJ, 1982, pp. 149–176.

[32] Dietterich, T., Lathrop, R., and Lozano-Perez, T. Solving the Multiple Instance Problem with Axis-Parallel Rectangles. *Artificial Intelligence 89*, 1–2 (January 1997), 31–71.

[33] Domingos, P., and Pazzani, M. Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. In *Proceedings of the Thirteenth International Conference on Machine Learning* (Bari, Italy, July 1996), pp. 105–112.

[34] Etzioni, O., Cafarella, M., Popescu, A., Shaked, T., Soderland, S., Weld, D., and Yates, A. Unsupervised Named-Entity Extraction from the Web: An Experimental Study. *Artificial Intelligence 165*, 1 (2005), 91–134.

[35] Fellbaum, C., Ed. *WordNet: An electronic lexical database*. MIT Press, 1998.

[36] Finkel, J., Grenager, T., and Manning, C. 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.

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

[38] Freitag, D. Information Extraction from HTML: Application of a General Machine Learning Approach. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence* (Madison, WI, July 1998), pp. 517–523.

[39] Freitag, D., and McCallum, A. 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.

[40] Gartner, T., Flach, P., Kowalczyk, A., and Smola, A. Multi-Instance Kernels. In *Proceedings of the 19th International Conference on Machine Learning* (Sydney, Australia, July 2002), pp. 179–186.

[41] Gee, F. R. The TIPSTER Text Program Overview. In *Proceedings of the TIPSTER Text Program Phase III* (Baltimore, MD, October 1998), pp. 3–5.

[42] Grenager, T., Klein, D., and Manning, C. Unsupervised Learning of Field Segmentation Models for Information Extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics* (Ann Arbor, MI, June 2005), pp. 371–378.

[43] Grishman, R., Huttunen, S., and Yangarber, R. Information Extraction for Enhanced Access to Disease Outbreak Reports. *Journal of Biomedical Informatics 35*, 4 (August 2002), 236–246.

[44] Grishman, R., Huttunen, S., and Yangarber, R. Real-Time Event Extraction for Infectious Disease Outbreaks. In *Proceedings of the 3rd Annual Human Language Technology Conference* (San Diego, CA, March 2002).

[45] Grishman, R., Sterling, J., and MacLeod, C. New York University PROTEUS System: MUC-3 Test Results and Analysis. In *Proceedings of the Third Message Understanding Conference (MUC-3)* (San Diego, CA, May 1991), pp. 95–98.

[46] Grishman, R., and Sundheim, B. Message Understanding Conference - 6: A Brief History. In *Proceedings of the 16th International Conference on Computational Linguistics* (Copenhagen, Denmark, August 1996), pp. 466–471.

[47] Gu, Z., and Cercone, N. 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.

[48] Harabagiu, S., Maiorano, S., and Pacsca, M. Open-domain Textual Question Answering Techniques. *Natural Language Engineering 9*, 3 (September 2003), 231–267.

[49] Hobbs, J. SRI International's TACITUS System: MUC-3 Test Results and Analysis. In *Proceedings of the Third Message Understanding Conference (MUC-3)* (San Diego, CA, May 1991), pp. 105–107.

[50] Hobbs, J., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M., and Tyson, M. FASTUS: A Cascaded Finite-state Transducer for Extracting Information for Natural-Language Text. In *Finite-State Language Processing*, E. Roche and Y. Schabes, Eds. MIT Press, Cambridge, MA, 1997, pp. 383–406.

[51] Hu, M., and Liu, B. Mining Opinion Features in Customer Reviews. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence* (San Jose, CA, July 2004), pp. 755–760.

[52] Huffman, S. 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, Berlin, 1996, pp. 246–260.

[53] Ittycheriah, A., Franz, M., and Roukos, S. IBM's Statistical Question Answering System – TREC-10. In *Proceedings of the Tenth Text REtrieval Conference (TREC-10)* (Gaithersburg, MD, November 2001), pp. 258–264.

[54] Ji, H., and Grishman, R. Improving Name Tagging by Reference Resolution and Relation Detection. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics* (Ann Arbor, MI, June 2005), pp. 411–418.

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

[56] Joachims, T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proceedings of the Tenth European Conference on Machine Learning* (April 1998), pp. 137–142.

[57] Kim, J., and Moldovan, D. PALKA: A System for Lexical Knowledge Acquisition. In *Proceedings of the Second International Conference on Information and Knowledge Management* (Washington, DC, November 1993), pp. 124–131.

[58] Koehn, P. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing* (Barcelona, Spain, July 2004), pp. 388–395.

[59] Krishnan, V., and Manning, C. An Effective Two-Stage Model for Exploiting Non-Local Dependencies in Named Entity Recognition. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics* (Sydney, Australia, July 2006), pp. 1121–1128.

[60] Krupka, G., Iwariska, L., Jacobs, P., and Rau, L. GE NLToolset: MUC-3 Test Results and Analysis. In *Proceedings of the Third Message Understanding Conference (MUC-3)* (San Diego, CA, May 1991), pp. 60–68.

[61] Landis, J., and Koch, G. The Measurement of Observer Agreement for Categorical Data. *Biometrics 33*, 1 (March 1977), 159–174.

[62] Lehnert, W., Cardie, C., Fisher, D., Riloff, E., and Williams, R. University of Massachusetts: MUC-3 Test Results and Analysis. In *Proceedings of the Third Message Understanding Conference (MUC-3)* (San Diego, CA, May 1991), pp. 116–119.

[63] Lehnert, W., and Cowie, J. Information Extraction. *Communications of the ACM 39*, 1 (January 1996), 80–91.

[64] Li, Y., Bontcheva, K., and Cunningham, H. 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.

[65] Light, M., Mann, G., Riloff, E., and Breck, E. Analyses for Elucidating Current Question Answering Technology. *Journal of Natural Language Engineering 7*, 4 (December 2001), 325–342.

[66] Lindberg, D., Humphreys, B., and McCray, A. The Unified Medical Language System. *Methods of Information in Medicine 32*, 4 (August 1993), 281–291.

[67] Liu, X., and Croft, B. Passage Retrieval Based On Language Models. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management* (McLean, VA, November 2002), pp. 375–382.

[68] Manning, C., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, 2008.

[69] Maslennikov, M., and Chua, T. A Multi-resolution Framework for Information Extraction from Free Text. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (Prague, Czech Republic, June 2007), pp. 592–599.

[70] Maslennikov, M., Goh, H., and Chua, T. ARE: Instance Splitting Strategies for Dependency Relation-based Information Extraction. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions* (Sydney, Australia, July 2006), pp. 571–578.

[71] McLernon, B., and Kushmerick, N. Transductive Pattern Learning for Information Extraction. In *Proceedings of the EACL 2006 Workshop on Adaptive Text Extraction and Mining* (Trento, Italy, April 2006), pp. 25–31.

[72] Melton, G., and Hripscak, G. Automated Detection of Adverse Events Using Natural Language Processing of Discharge Summaries. *Journal of the American Medical Informatics Association 12*, 4 (July 2005), 448–457.

[73] Merchant, R. TIPSTER Program Overview. In *TIPSTER Text Program Phase I: Proceedings of the Workshop* (Fredricksburg, VA, September 1993), pp. 1–2.

[74] Mihalcea, R. Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization. In *The Companion Volume to the Proceedings of 42st Annual Meeting of the Association for Computational Linguistics* (Barcelona, Spain, July 2004), pp. 170–173.

[75] Mikheev, A., Moens, M., and Grover, C. Named Entity Recognition without Gazetteers. In *Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics* (Bergen, Norway, June 1999), pp. 1–8.

[76] Mitchell, T. *Machine Learning*. McGraw–Hill, Boston, MA, 1997.

[77] Mittendorf, E., and Schaüble, P. Document and Passage Retrieval Based on Hidden Markov Models. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Dublin, Ireland, July 1994), pp. 318–327.

[78] Nomoto, T., and Matsumoto, Y. A New Approach to Unsupervised Text Summarization. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New Orleans, LA, September 2001), pp. 26–34.

[79] Oliva, A., and Torralba, A. The Role of Context in Object Recognition. *Trends in Cognitive Sciences 11*, 12 (December 2007), 520–527.

[80] Pang, B., Lee, L., and Vaithyanathan, S. Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Philadelphia, PA, July 2002), pp. 79–86.

[81] Patwardhan, S., and Riloff, E. Learning Domain-Specific Information Extraction Patterns from the Web. In *Proceedings of the ACL 2006 Workshop on Information Extraction Beyond the Document* (Sydney, Australia, July 2006), pp. 66–73.

[82] Patwardhan, S., and Riloff, E. Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (Prague, Czech Republic, June 2007), pp. 717–727.

[83] Patwardhan, S., and Riloff, E. A Unified Model of Phrasal and Sentential Evidence for Information Extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (Singapore, August 2009), pp. 151–160.

[84] Peng, F., and McCallum, A. Accurate Information Extraction from Research Papers using Conditional Random Fields. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (Boston, MA, May 2004), pp. 329–336.

[85] Penz, J., Wilcox, A., and Hurdle, J. Automated Identification of Adverse Events Related to Central Venous Catheters. *Journal of Biomedical Informatics 40*, 2 (April 2007), 174–182.

[86] Phillips, W., and Riloff, E. Exploiting Role-Identifying Nouns and Expressions for Information Extraction. In *Proceedings of International Conference on Recent Advances in Natural Language Processing* (Borovets, Bulgaria, September 2007), pp. 165–172.

[87] Poibeau, T., and Dutoit, D. Generating Extraction Patterns from a Large Semantic Network and an Untagged Corpus. In *Proceedings of the COLING-02 Workshop on SEMANET: Building and Using Semantic Networks* (Taipei, Taiwan, August 2002), pp. 1–7.

[88] Popescu, A., and Etzioni, O. Extracting Product Features and Opinions from Reviews. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (Vancouver, Canada, October 2005), pp. 339–346.

[89] Prager, J., Brown, E., Coden, A., and Radev, D. Question Answering by Predictive Annotation. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Athens, Greece, July 2000), pp. 184–191.

[90] Ravichandran, D., and Hovy, E. Learning Surface Text Patterns for a Question Answering System. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (Philadelphia, PA, July 2002), pp. 41–47.

[91] Riloff, E. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence* (Washington, DC, July 1993), pp. 811–816.

[92] Riloff, E. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of the Thirteenth National Conference on Articial Intelligence* (Portland, OR, August 1996), pp. 1044–1049.

[93] Riloff, E., Mann, G., and Phillips, W. Reverse-Engineering Question/Answer Collections From Ordinary Text. In *Advances in Open Domain Question Answering*, T. Strzalkowski and S. Harabagiu, Eds. Springer, The Netherlands, 2006, pp. 505–531.

[94] Riloff, E., Patwardhan, S., and Wiebe, J. Feature Subsumption for Opinion Analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (Sydney, Australia, July 2006), pp. 440–448.

[95] Riloff, E., and Phillips, W. An Introduction to the Sundance and AutoSlog Systems. Tech. Rep. UUCS-04-015, School of Computing, University of Utah, 2004.

[96] Riloff, E., Wiebe, J., and Phillips, W. Exploiting Subjectivity Classification to Improve Information Extraction. In *Proceedings of the Twentieth National Conference on Artificial Intelligence* (Pittsburgh, PA, July 2005), pp. 1106–1111.

[97] Roth, D., and Yih, W. 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.

[98] Sager, N. *Natural Language Information Processing: A Computer Grammar of English and Its Applications*. Addison-Wesley, Boston, MA, 1981.

[99] Salton, G., Allan, J., and Buckley, C. Approaches to Passage Retrieval in Full Text Information Systems. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval* (Pittsburgh, PA, June 1993), pp. 49–58.

[100] Schank, R., and Abelson, R. *Scripts, Plans, and Understanding*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.

[101] Sekine, S. On-Demand Information Extraction. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions* (Sydney, Australia, July 2006), pp. 731–738.

[102] Shinyama, Y., and Sekine, S. 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, NY, June 2006), pp. 304–311.

[103] Silva, G., and Dwiggins, D. Towards a Prolog Text Grammar. *ACM SIGART Bulletin 73* (October 1980), 20–25.

[104] Smith, A., and Osborne, M. Using Gazetteers in Discriminative Information Extraction. In *Proceedings of the 10th Conference on Computational Natural Language Learning* (New York, NY, June 2006), pp. 133–140.

[105] Soderland, S. Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning 34*, 1-3 (February 1999), 233–272.

[106] Soderland, S., Fisher, D., Aseltine, J., and Lehnert, W. CRYSTAL: Inducing a Conceptual Dictionary. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (Montreal, Canada, August 1995), pp. 1314–1319.

[107] Soricut, R., and Marcu, D. Sentence Level Discourse Parsing using Syntactic and Lexical Information. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (Edmonton, Canada, May 2003), pp. 149–156.

[108] Srihari, R., and Li, W. A Question Answering System Supported by Information Extraction. In *Proceedings of the Sixth Applied Natural Language Processing Conference* (Seattle, WA, April 2000), pp. 166–172.

[109] Srihari, R., Li, W., and Li, X. Question Answering Supported by Multiple Levels of Information Extraction. In *Advances in Open Domain Question Answering*, T. Strzalkowski and S. Harabagiu, Eds. Springer, The Netherlands, 2006, pp. 349–382.

[110] Stevenson, M., and Greenwood, M. 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] Sudo, K., Sekine, S., and Grishman, R. An Improved Extraction Patterns Representation Model for Automatic IE Pattern Acquisition. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (Sapporo, Japan, July 2003), pp. 224–231.

[112] Sun, R., Ong, C., and Chua, T. Mining Dependency Relations for Query Expansion in Passage Retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Seattle, WA, August 2006), pp. 382–389.

[113] Sundheim, B. Overview of the Third Message Understanding Evaluation and Conference. In *Proceedings of the Third Message Understanding Conference (MUC-3)* (San Diego, CA, May 1991), pp. 3–16.

[114] Sundheim, B. Overview of the Fourth Message Understanding Evaluation and Conference. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)* (McLean, VA, June 1992), pp. 3–21.

[115] Sundheim, B. Overview of the Results of the MUC-6 Evaluation. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)* (Los Altos, CA, November 1995), pp. 13–31.

[116] Surdeanu, M., Harabagiu, S., Williams, J., and Aarseth, P. Using Predicate-Argument Structures for Information Extraction. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics* (Sapporo, Japan, July 2003), pp. 8–15.

[117] Tellex, S., Katz, B., Lin, J., Fernandes, A., and Marton, G. Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Toronto, Canada, July 2003), pp. 41–47.

[118] Torralba, A., Murphy, K., Freeman, W., and Rubin, M. Context-Based Vision System for Place and Object Recognition. In *IEEE International Conference on Computer Vision* (Nice, France, October 2003), pp. 273–280.

[119] Vapnik, V. *The Nature of Statistical Learning Theory.* Springer, New York, NY, 1995.

[120] Wiebe, J., and Riloff, E. Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. In *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing* (Mexico City, Mexico, February 2005), pp. 486–497.

[121] Wiebe, J., Wilson, T., and Cardie, C. Annotating Expressions of Opinions and Emotions in Language. *Language Resources and Evaluation 39*, 2-3 (May 2005), 165–210.

[122] Witten, I., and Frank, E. *Data Mining - Practical Machine Learning Tools and Techniques.* Morgan–Kaufmann, San Francisco, CA, 2005.

[123] Xiao, J., Chua, T., and Cui, H. Cascading Use of Soft and Hard Matching Pattern Rules for Weakly Supervised Information Extraction. In *Proceedings of the 20th International Conference on Computational Linguistics* (Geneva, Switzerland, August 2004), pp. 542–548.

[124] Yangarber, R., Grishman, R., Tapanainen, P., and Huttunen, S. Automatic Acquisition of Domain Knowledge for Information Extraction. In *Proceedings of the 18th International Conference on Computational Linguistics* (Saarbrücken, Germany, August 2000), pp. 940–946.

[125] Yangarber, R., Lin, W., and Grishman, R. Unsupervised Learning of Generalized Names. In *Proceedings of the 19th International Conference on Computational Linguistics* (Taipei, Taiwan, August 2002), pp. 154–160.

[126] Yu, K., Guan, G., and Zhou, M. 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.

[127] Zadrozny, B., and Elkan, C. Obtaining Calibrated Probability Estimates from Decision Trees and Naïve Bayesian Classiers. In *Proceedings of the Eighteenth International Conference on Machine Learning* (Williamstown, MA, June 2001), pp. 609–616.

[128] Zarri, G. Automatic Representation of the Semantic Relationships Corresponding to a French Surface Expression. In *Proceedings of the First Conference on Applied Natural Language Processing* (Santa Monica, CA, February 1983), pp. 143–147.

[129] Zelenko, D., Aone, C., and Richardella, A. Kernel Methods for Relation Extraction. *Journal of Machine Learning Research 3* (February 2003), 1083–1106.

[130] Zhou, G., and Zhang, M. Extracting Relation Information from Text Documents by Exploring Various Types of Knowledge. *Information Processing and Management 43*, 4 (July 2007), 969–982.