Chapter 21

Information Extraction

Jerry R. Hobbs, University of Southern California
Ellen Riloff, University of Utah

Abstract *Information Extraction* (IE) techniques aim to extract the names of entities and objects from text and to identify the roles that they play in event descriptions. IE systems generally focus on a specific domain or topic, searching only for information that is relevant to a user's interests. In this chapter, we first give historical background on information extraction and discuss several kinds of information extraction tasks that have emerged in recent years. Next, we outline the series of steps that are involved in creating a typical information extraction system, which can be encoded as a cascaded finite-state transducer. Along the way, we present examples to illustrate what each step does. Finally, we present an overview of different learning-based methods for information extraction, including supervised learning approaches, weakly supervised and bootstrapping techniques, and discourse-oriented approaches.

21.1 Introduction

Information extraction (IE) is the process of scanning text for information relevant to some interest, including extracting entities, relations, and, most challenging, events–or who did what to whom when and where. It requires deeper analysis than key word searches, but its aims fall short of the very hard and long-term problem of text understanding, where we seek to capture all the information in a text, along with the speaker's or writer's intention.

Information extraction represents a midpoint on this spectrum, where the aim is to capture structured information without sacrificing feasibility. IE typically focuses on surface linguistic phenomena that do not require deep inference, and it focuses on the phenomena that are most frequent in texts.

Information extraction technology arose in response to the need for efficient processing of texts in specialized domains. Full-sentence parsers expended a lot of effort in trying to arrive at parses of long sentences that were not relevant to the domain, or which contained much irrelevant material, thereby increasing the chances for error. Information extraction technology, by contrast, focuses in on only the relevant parts of the text and ignores the rest.

Typical applications of information extraction systems are in gleaning business, government, or military intelligence from a large number of sources; in searches of the World Wide Web for more specific information than keywords can discriminate; for scientific literature searches; in building databases from large textual corpora; and in the curation of biomedical articles. The need for information extraction is well illustrated in biomedicine, where there are more than half a million articles a year, and large amounts of money are spent on curatorial activities. Similarly, in intelligence gathering, an analyst in 1990 said that reading everything she was supposed to read would be like reading *War and Peace* every day; in 1995 the same analyst said it was way beyond that.

Named entity recognition (NER) is one of the most common uses of information extraction technology (e.g., (Bikel, Schwartz, and Weischedel 1999; Collins and Singer 1999; Cucerzan and Yarowsky 1999; Fleischman and Hovy 2002; Sang and Meulder 2003)).

NER systems identify different types of proper names, such as person and company names, and sometimes special types of entities, such as dates and times, that can be easily identified using surface-level textual patterns. NER is especially important in biomedical applications, where terminology is a formidable problem. But it is important to note that information extraction is much more than just named entity recognition. A much more difficult and potentially much more significant capability is the recognition of events and their participants. For example, in each of the sentences:

> "Microsoft acquired Powerset."
> "Powerset was acquired by Microsoft."

we would like to recognize not only that Microsoft and Powerset are company names, but also that an acquisition event took place, that the acquiring company was Microsoft, and the acquired company was Powerset.

Much of the technology in information extraction was developed in response to a series of evaluations and associated conferences called the Message Understanding Conference (MUC), held between 1987 and 1998.

Except for the earliest MUCs, these evaluations were based on a corpus of domain-specific texts, such as news articles on joint ventures. Participating teams were supplied with a training corpus and a template definition for

the events and their roles. For joint ventures, the roles were such things as the participating companies, the joint venture company that was formed, the activity it would engage in, and the amount of money it was capitalized for. The systems were then run on a previously unseen test corpus. A system's performance was measured on recall (what percentage of the correct answers did the system get), precision (what percentage of the system's answers were correct), and F-score. F-score is a a weighted harmonic mean between recall and precision computed by the following formula:

$$F = \frac{(\beta^2+1)PR}{\beta^2 P+R}$$

where P is precision, R is recall, and β is a parameter encoding the relative importance of recall and precision. If $\beta = 1$, they are weighted equally. If $\beta > 1$, precision is more important; if $\beta < 1$, recall is more important.[1]

A typical text in the joint ventures domain used in MUC-5 (July 1993) (MUC-5 Proceedings 1993) is the following:

> Bridgestone Sports Co. said Friday it has set up a joint venture in Taiwan with a local concern and a Japanese trading house to produce golf clubs to be shipped to Japan.
>
> The joint venture, Bridgestone Sports Taiwan Co., capitalized at 20 million new Taiwan dollars, will start production in January 1990 with production of 20,000 iron and "metal wood" clubs a month.

The information to be extracted from this text is shown in the following templates:

TIE-UP-1:	
Relationship:	TIE-UP
Entities:	"Bridgestone Sports Co."
	"a local concern"
	"a Japanese trading house"
Joint Venture Company:	"Bridgestone Sports Taiwan Co."
Activity:	ACTIVITY-1
Amount:	NT$20000000

ACTIVITY-1:	
Activity:	PRODUCTION
Company:	"Bridgestone Sports Taiwan Co."
Product:	"iron and 'metal wood' clubs"
Start Date:	DURING: January 1990

[1] When in a courtroom you promise to tell the whole truth, you are promising 100% recall. When you promise to tell nothing but the truth, you are promising 100% precision.

IE research has since been stimulated by the Automatic Content Extraction (ACE) evaluations[2]. The ACE evaluations have focused on identifying named entities, extracting isolated relations, and coreference resolution.

Information extraction systems have been developed for a variety of domains, including terrorist events (MUC-4 Proceedings 1992; Chieu, Ng, and Lee 2003; Riloff 1996b; Soderland, Fisher, Aseltine, and Lehnert 1995), joint ventures (MUC-5 Proceedings 1993), management succession (MUC-6 Proceedings 1995), plane crashes (MUC-7 Proceedings 1998), vehicle launches (MUC-7 Proceedings 1998), corporate acquisitions (Freitag 1998b; Freitag and McCallum 2000; Finn and Kushmerick 2004), disease outbreaks (Grishman, Huttunen, and Yangarber 2002; Patwardhan and Riloff 2007; Phillips and Riloff 2007), job postings (Califf and Mooney 2003; Freitag and McCallum 2000), rental ads (Soderland 1999; Ciravegna 2001), resumes (Yu, Guan, and Zhou 2005), and seminar announcements (Freitag 1998b; Ciravegna 2001; Chieu and Ng 2002; Califf and Mooney 2003; Finn and Kushmerick 2004; Gu and Cercone 2006), There has also been a great deal of work on information extraction in biological and medical domains (e.g., (Friedman 1986; Subramaniam, Mukherjea, Kankar, Srivastava, Batra, Kamesam, and Kothari 2003; Ananiadou, Friedman, and Tsujii 2004; Hirschman, Yeh, Blaschke, and Valencia 2005; Yakushiji, Miyao, Ohta, and Tateisi 2006; Ananiadou and McNaught 2006)), which is discussed in greater depth in the BioNLP chapter of this book.

21.2 Diversity of IE Tasks

The Message Understanding Conferences led to an increased interest in the IE task and the creation of additional IE data sets. Researchers began to work on IE problems for new domains and focused on different aspects of the information extraction problem. In the following sections, we outline some of the fundamental distinctions that cut across different information extraction tasks.

21.2.1 Unstructured vs. Semi-Structured Text

Historically, most natural language processing systems have been designed to process *unstructured text*, which consists of natural language sentences. In contrast to structured data where the semantics of the data is defined by its organization (e.g., database entries), the meaning of unstructured text depends entirely on linguistic analysis and natural language understanding.

[2] http://www.itl.nist.gov/iad/mig/tests/ace/

Professor John Skvoretz, U. of South Carolina, Columbia, will present a seminar entitled "Embedded Commitment," on Thursday, May 4th from 4-5:30 in PH 223D.

FIGURE 21.1: Example of an unstructured seminar announcement

Examples of unstructured text include news stories, magazine articles, and books.[3] Figure 21.1 shows an example of a seminar announcement that is written as unstructured text.

Semi-structured text consists of natural language that appears in a document where the physical layout of the language plays a role in its intepretation. For example, consider the seminar announcements depicted in Figure 21.2. The reader understands that the speaker is Laura Petitte, who is from the Department of Psychology at McGill University, because seminar speakers and their affiliations typically appear at the top of a seminar announcement. If McGill University had appeared below Baker Hall 355 in the announcement, then we would assume that the seminar takes place at McGill University.

Several IE data sets have been created specifically to handle domains that often include semi-structured text, such as seminar announcements, job postings, rental ads, and resumes. To accommodate semi-structured text, IE systems typically rely less on syntactic parsing and more on positional features that capture the physical layout of the words on the page.

21.2.2 Single-Document vs. Multi-Document IE

Originally, information extraction systems were designed to locate domain-specific information in individual documents. Given a document as input, the IE system identifies and extracts facts relevant to the domain that appear in the document. We will refer to this task as *single-document information extraction*.

The abundance of information available on the Web has led to the creation of new types of IE systems that seek to extract facts from the Web or other very large text collections (e.g., (Brin 1998; Fleischman, Hovy, and Echihabi 2003; Etzioni, Cafarella, Popescu, Shaked, Soderland, Weld, and Yates 2005; Pasca, Lin, Bigham, Lifchits, and Jain 2006; Pasca 2007; Banko, Cafarella, Soderland, Broadhead, and Etzioni 2007)). We will refer to this task as *multi-document information extraction*

Single-document IE is fundamentally different from multi-document IE, although both types of systems may use similar techniques. One distinguishing issue is redundancy. A single-document IE system must extract domain-

[3]These text forms can include some structured information as well, such as publication dates and author by-lines. But most of the text in these genres is unstructured.

Laura Petitte
Department of Psychology
McGill University

Thursday, May 4, 1995
12:00 pm
Baker Hall 355

Name: Dr. Jeffrey D. Hermes
Affiliation: Department of AutoImmune Diseases
Research & Biophysical Chemistry Merck Research Laboratories
Title: "MHC Class II: A Target for Specific Immunomodulation of the Immune Response"
Host/e-mail: Robert Murphy, murph@a.crf.cmu.edu
Date: Wednesday, May 3, 1995
Time: 3:30 p.m.
Place: Mellon Institute Conference Room
Sponsor: MERCK RESEARCH LABORATORIES

FIGURE 21.2: Examples of semi-structured seminar announcements

specific information from each document that it is given. If the system fails to find relevant information in a document, then that is an error. This task is challenging because many documents mention a fact only once, and the fact may be expressed in an unusual or complex linguistic context (e.g., one requiring inference). In contrast, multi-document IE systems can exploit the redundancy of information in its large text collection. Many facts will appear in a wide variety of contexts, so the system usually has multiple opportunities to find each piece of information. The more often a fact appears, the greater the chance that it will occur at least once in a linguistically simple context that will be straightforward for the IE system to recognize.[4]

Multi-document IE is sometimes referred to as "open-domain" IE because the goal is usually to acquire broad-coverage factual information, which will likely benefit many domains. In this paradigm, it doesn't matter where the information originated. Some open-domain IE systems, such as KnowItAll (Etzioni, Cafarella, Popescu, Shaked, Soderland, Weld, and Yates 2005) and TextRunner (Banko, Cafarella, Soderland, Broadhead, and Etzioni 2007), have

[4]This issue parallels the difference between single-document and multi-document question answering (QA) systems. Light et al. (Light, Mann, Riloff, and Breck 2001) found that the performance of QA systems in TREC-8 was directly correlated with the number of answer opportunities available for a question.

addressed issues of scale to acquire large amounts of information from the Web. One of the major challenges in multi-document IE is cross-document coreference resolution: when are two documents talking about the same entities? Some researchers have tackled this problem (e.g., (Bagga and Baldwin 1998; Mann and Yarowsky 2003; Gooi and Allan 2004; Niu, Li, and Srihari 2004; Mayfield, Alexander, Dorr, Eisner, Elsayed, Finin, Fink, Freedman, Garera, McNamee, Mohammad, Oard, Piatko, Sayeed, Syed, Weischedel, Xu, and Yarowsky 2009)), and in 2008 the ACE evaluation expanded its focus to include cross-document entity disambiguation (Strassel, Przybocki, Peterson, Song, and Maeda 2008).

21.2.3 Asssumptions about Incoming Documents

The IE data sets used in the Message Understanding Conferences consist of documents related to the domain, but not all of the documents mention a relevant event. The data sets were constructed to mimic the challenges that a real-world information extraction system must face, where a fundamental part of the IE task is to determine whether a document describes a relevant event, as well as to extract information about the event. In the MUC-3 through MUC-7 IE data sets, only about half of the documents describe a domain-relevant event that warrants information extraction.

Other IE data sets make different assumptions about the incoming documents. Many IE data sets consist only of documents that describe a relevant event. Consequently, the IE system can assume that each document contains information that should be extracted. This assumption of *relevant-only documents* allows an IE system to be more aggressive about extracting information because the texts are known to be on-topic. For example, if an IE system is given stories about bombing incidents, then it can extract the name of every person who was killed or injured and in most cases they will be victims of a bombing. If, however, irrelevant stories are also given to the system, then it must further distinguish between people who are bombing victims and people who were killed or injured in other types of events, such as robberies or car crashes.

Some IE data sets further make the assumption that each incoming document contains only one event of interest. We will refer to these as *single-event documents*. The seminar announcements, corporate acquisitions, and job postings IE data sets only contain single-event documents. In contrast, the MUC data sets and some others (e.g., rental ads and disease outbreaks) allow that a single document may describe multiple events of interest. If the IE system can assume that each incoming document describes only one relevant event, then all of the extracted information can be inserted in a single output template.[5] If multiple events are discussed in a document, then the

[5]Note that coreference resolution of entities is still an issue, however. For example, a

IE system must perform discourse analysis to determine how many different events are being reported and to associate each piece of extracted information with the appropriate event template.

21.3 IE with Cascaded Finite-State Transducers

Probably the most important idea that emerged in the course of the MUC evaluations was the decomposition of the IE process into a series of subproblems that can be modeled with "cascaded finite-state transducers" (Lehnert, Cardie, Fisher, Riloff, and Williams 1991; Hobbs, Appelt, Bear, Israel, and Tyson 1992; Hobbs, Appelt, Bear, Israel, Kameyama, Stickel, and Tyson 1997; Joshi 1996; Cunningham, Maynard, Bontcheva, and Tablan 2002). A finite-state automaton reads one element at a time of a sequence of elements; each element transitions the automaton into a new state, based on the type of element it is, e.g., the part of speech of a word. Some states are designated as final, and a final state is reached when the sequence of elements matches a valid pattern. In a finite-state transducer, an output entity is constructed when final states are reached, e.g., a representation of the information in a phrase. In a cascaded finite-state transducer, there are different finite-state transducers at different stages. Earlier stages will package a string of elements into something the next stage will view as a single element.

In the typical system, the earlier stages recognize smaller linguistic objects and work in a largely domain-independent fashion. They use purely linguistic knowledge to recognize portions of the syntactic structure of a sentence that linguistic methods can determine reliably, requiring relatively little modification or augmentation as the system is moved from domain to domain. The later stages take these linguistic objects as input and find domain-dependent patterns within them. In a typical IE system, there are five levels of processing:

1. Complex Words: This includes the recognition of multiwords and proper name entities, such as people, companies, and countries.

2. Basic Phrases: Sentences are segmented into noun groups, verb groups, and particles.

3. Complex Phrases: Complex noun groups and complex verb groups are identified.

document may mention multiple victims so the IE system needs to determine whether an extracted victim refers to a previously mentioned victim or a new one.

4. Domain Events: The sequence of phrases produced at Level 3 is scanned for patterns of interest to the application, and when they are found, semantic structures are built that encode the information about entities and events contained in the pattern.

5. Merging Structures: Semantic structures from different parts of the text are merged if they provide information about the same entity or event. This process is sometimes called *template generation*, and is a complex process not done by a finite-state transducer.

As we progress through the five levels, larger segments of text are analyzed and structured. In each of stages 2 through 4, the input to the finite-state transducer is the sequence of chunks constructed in the previous stage. The GATE project (Cunningham, Maynard, Bontcheva, and Tablan 2002) is a widely used toolkit that provides many of the components needed for such an IE pipeline.

This decomposition of the natural-language problem into levels is essential to the approach. Many systems have been built to do pattern matching on strings of words. The advances in information extraction have depended crucially on dividing that process into separate levels for recognizing phrases and recognizing patterns among the phrases. Phrases can be recognized reliably with purely syntactic information, and they provide precisely the elements that are required for stating the patterns of interest.

In the next five sections we illustrate this process on the Bridgestone Sports text.

21.3.1 Complex Words

The first level of processing identifies multiwords such as "set up", "trading house", "new Taiwan dollars", and "joint venture", and company names like "Bridgestone Sports Co." and "Bridgestone Sports Taiwan Co.". The names of people and locations, dates, times, and other basic entities are also recognized at this level. Languages in general are very productive in the construction of short, multiword fixed phrases and proper names employing specialized microgrammars, and this is the level at which they are recognized.

Some names can be recognized by their internal structure. A common pattern for company names is "ProperName ProductName", as in "Acme Widgets". Others can only be recognized by means of a table. Internal structure cannot tell us that IBM is a company and DNA is not. It is also sometimes possible to recognize the types of proper names by the context in which they occur. For example, in the sentences below:

(a) XYZ's sales
(b) Vaclav Havel, 53, president of the Czech Republic

we might not know that XYZ is a company and Vaclav Havel is a person, but

the immediate context establishes that. These can be given an underspecified representation that is resolved by later stages.

21.3.2 Basic Phrases

The problem of syntactic ambiguity in natural language is AI-complete. That is, we will not have systems that reliably parse English sentences correctly until we have encoded much of the real-world knowledge that people bring to bear in their language comprehension. For example, noun phrases cannot be reliably identified because of the prepositional phrase attachment problem. However, certain syntactic constructs can be identified with reasonable reliability. One of these is the noun group, which is the head noun of a noun phrase together with its determiners and other left modifiers (these are sometimes called "base NPs"). Another is what we are calling the "verb group", that is, the verb together with its auxiliaries and any intervening adverbs. Moreover, an analysis that identifies these elements gives us exactly the units we most need for subsequent domain-dependent processing. The task of identifying these simple noun and verb groups is sometimes called "syntactic chunking". The basic phrases in the first sentence of text (1) are as follows, where "Company Name" and "Location" are special kinds of noun group that would be identified by named entity recognition:

Company Name:	Bridgestone Sports Co.
Verb Group:	said
Noun Group:	Friday
Noun Group:	it
Verb Group:	had set up
Noun Group:	a joint venture
Preposition:	in
Location:	Taiwan
Preposition:	with
Noun Group:	a local concern
Conjunction:	and
Noun Group:	a Japanese trading house
Verb Group:	to produce
Noun Group:	golf clubs
Verb Group:	to be shipped
Preposition:	to
Location:	Japan

Noun groups can be recognized by a relatively simple finite-state grammar encompassing most of the complexity that can occur in English noun groups (Hobbs et al., 1992), including numbers, numerical modifiers like "approximately", other quantifiers and determiners, participles in adjectival position, comparative and superlative adjectives, conjoined adjectives, and arbitrary or-

derings and conjunctions of prenominal nouns and noun-like adjectives. Thus, among the noun groups that can be recognized are:

"approximately 5 kg"
"more than 30 people"
"the newly elected president"
"the largest leftist political force"
"a government and commercial project"

The principal ambiguities that arise in this stage are due to noun-verb ambiguities. For example, "the company names" could be a single noun group with the head noun "names", or it could be a noun group "the company" followed by the verb "names". One can use a lattice representation to encode the two analyses and resolve the ambiguity in the stage for recognizing domain events.

Verb groups (and predicate adjective constructions) can be recognized by an even simpler finite-state grammar that, in addition to chunking, also tags them as Active Voice, Passive Voice, Gerund, and Infinitive. Verbs are sometimes locally ambiguous between active and passive senses, as the verb "kidnapped" in the following two sentences:

"Several men kidnapped the mayor today."
"Several men kidnapped yesterday were released today."

These cases can be tagged as Active/Passive, and the domain-event stage can later resolve the ambiguity. Some work has also been done to train a classifier to distinguish between active voice and "reduced" passive voice constructions (Igo and Riloff 2008).

The breakdown of phrases into nominals, verbals, and particles is a linguistic universal. Whereas the precise parts of speech that occur in any language can vary widely, every language has elements that are fundamentally nominal in character, elements that are fundamentally verbal or predicative, and particles or inflectional affixes that encode relations among the other elements (Croft 1991).

21.3.3 Complex Phrases

Some complex noun groups and verb groups can be recognized reliably on the basis of domain-independent, syntactic information. For example:

- the attachment of appositives to their head noun group

  "The joint venture, Bridgestone Sports Taiwan Co.,"

- the construction of measure phrases

  "20,000 iron and "metal wood" clubs a month"

- the attachment of "of" and "for" prepositional phrases to their head noun groups

"production of 20,000 iron and "metal wood" clubs a month"

- noun group conjunction

"a local concern and a Japanese trading house"

In the course of recognizing basic and complex phrases, domain-relevant entities and events can be recognized and the structures for these can be constructed. In the sample joint-venture text, entity structures can be constructed for the companies referred to by the phrases "Bridgestone Sports Co.", "a local concern", "a Japanese trading house", and "Bridgestone Sports Taiwan Co." Information about nationality derived from the words "local" and "Japanese" can be recorded. Corresponding to the complex noun group "The joint venture, Bridgestone Sports Taiwan Co.," the following relationship structure can be built:

Relationship:	TIE-UP
Entities:	–
Joint Venture Company:	"Bridgestone Sports Taiwan Co."
Activity:	–
Amount:	–

Corresponding to the complex noun group "production of 20,000 iron and 'metal wood' clubs a month", the following activity structure can be built up:

Activity:	PRODUCTION
Company:	–
Product:	"iron and 'metal wood' clubs"
Start Date:	–

Complex verb groups can also be recognized in this stage. Consider the following variations:

"GM *formed* a joint venture with Toyota."
"GM *announced it was forming* a joint venture with Toyota."
"GM *signed an agreement forming* a joint venture with Toyota."
"GM *announced it was signing an agreement to form* a joint venture with Toyota."

Although these sentences may differ in significance for some applications, often they would be considered equivalent in meaning. Rather than defining each of these variations, with all their syntactic variants, at the domain event level, the user should be able to define complex verb groups that share the same significance. Thus, "formed", "announced it was forming", "signed an agreement forming", and "announced it was signing an agreement to form" may all be equivalent, and once they are defined to be so, only one domain event pattern needs to be expressed. Verb group conjunction, as in

"Terrorists *kidnapped and killed* three people."

can be treated as a complex verb group as well.

21.3.4 Domain Events

The next stage is recognizing domain events, and its input is a a list of the basic and complex phrases recognized in the earlier stages, in the order in which they occur. Anything that was not identified as a basic or complex phrase in a previous stage can be ignored in this stage; this can be a significant source of robustness.

Identifying domain events requires a set of domain-specific patterns both to recognize phrases that correspond to an event of interest and to identify the syntactic constitutents that correspond to the event's role fillers. In early information systems, these domain-specific "extraction patterns" were defined manually. In Sections 21.4.1 and 21.4.3, we describe a variety of learning methods that have subsequently been developed to automatically generate domain-specific extraction patterns from training corpora.

The patterns for events of interest can be encoded as finite-state machines, where state transitions are effected by phrases. The state transitions are driven off the head words in the phrases. That is, each pair of relevant head word and phrase type—such as "company-NounGroup" and "formed-PassiveVerbGroup"— has an associated set of state transitions. In the sample joint-venture text, the domain event patterns

<Company/ies> <Set-up> <Joint-Venture> with <Company/ies>

and

<Produce> <Product>

would be instantiated in the first sentence, and the patterns

<Company> <Capitalized> at <Currency>

and

<Company> <Start> <Activity> in/on <Date>

in the second. These four patterns would result in the following four structures being built:

Relationship:	TIE-UP
Entities:	"Bridgestone Sports Co."
	"a local concern"
	"a Japanese trading house"
Joint Venture Company:	–
Activity:	–
Amount:	–

Activity:	PRODUCTION
Company:	–
Product:	"golf clubs"
Start Date:	–

Relationship:	TIE-UP
Entities:	–
Joint Venture Company:	"Bridgestone Sports Taiwan Co."
Activity:	–
Amount:	NT$20000000

Activity:	PRODUCTION
Company:	"Bridgestone Sports Taiwan Co."
Product:	–
Start Date:	DURING: January 1990

The third of these is an augmentation of the TIE-UP structure discovered in the complex phrase phase.

Certain kinds of "pseudo-syntax" can be done at this stage, including recognizing relative clauses and conjoined verb phrases, as described in Hobbs et al. (1997).

Many subject-verb-object patterns are of course related to each other. The sentence:

"GM manufactures cars."

illustrates a general pattern for recognizing a company's activities. But the same semantic content can appear in a variety of ways, including

"Cars are manufactured by GM."
"... GM, which manufactures cars..."
"... cars, which are manufactured by GM..."
"... cars manufactured by GM ..."
"GM is to manufacture cars."
"Cars are to be manufactured by GM."
"GM is a car manufacturer."

These are all systematically related to the active voice form of the sentence. Therefore, there is no reason a developer should have to specify all the variations. A simple tool would be able to generate all of the variants of the pattern from the simple active voice Subject-Verb-Object form. It would also allow adverbials to appear at appropriate points. These transformations would be executed at compile time, producing the more detailed set of patterns, so that at run time there is no loss of efficiency.

This feature is not merely a clever idea for making a system more convenient to author. It rests on the fundamental idea that underlies generative transformational grammar, but is realized in a way that does not impact the efficiency of processing.

In recent years, full-sentence parsing has improved, in large part through the use of statistical techniques. Consequently, some IE systems have begun to rely on full parsers rather than shallow parsing techniques.

21.3.5 Template Generation: Merging Structures

The first four stages of processing all operate within the bounds of single sentences. The final level of processing operates over the whole text. Its task is to see that all the information collected about a single entity, relationship, or event is combined into a unified whole. This is one of the primary ways that the problem of coreference is dealt with in information extraction, including both NP coreference (for entities) and event coreference. One event template is generated for each event, which coalesces all of the information associated with that event. If an input document discusses multiple events of interest, then the IE system must generate multiple event templates. Generating multiple event templates requires additional discourse analysis to (a) correctly determine how many distinct events are reported in the document, and (b) correctly assign each entity and object to the appropriate event template.

Among the criteria that need to be taken into account in determining whether two structures can be merged are the internal structure of the noun groups, nearness along some metric, and the consistency, or more generally, the compatibility of the two structures.

In the analysis of the sample joint-venture text, we have produced three activity structures. They are all consistent because they are all of type PRODUCTION and because "iron and 'metal wood' clubs" is consistent with "golf clubs". Hence, they are merged, yielding:

Activity:	PRODUCTION
Company:	"Bridgestone Sports Taiwan Co."
Product:	"iron and 'metal wood' clubs"
Start Date:	DURING: January 1990

Similarly, the two relationship structures that have been generated are consistent with each other, so they can be merged, yielding:

Relationship:	TIE-UP
Entities:	"Bridgestone Sports Co."
	"a local concern"
	"a Japanese trading house"
Joint Venture Company:	"Bridgestone Sports Taiwan Co."
Activity:	–
Amount:	NT$20000000

The entity and event coreference problems are very hard, and constitute active and important areas of research. Coreference resolution was a task in the later MUC evaluations (MUC-6 Proceedings 1995; MUC-7 Proceedings 1998), and has been a focus of the ACE evaluations. Many recent research efforts have applied machine learning techniques to the problem of coreference resolution (e.g., (Dagan and Itai 1990; McCarthy and Lehnert 1995; Aone and Bennett 1996; Kehler 1997; Cardie and Wagstaff 1999; Harabagiu, Bunescu, and Maiorana 2001; Soon, Ng, and Lim 2001; Ng and Cardie 2002; Bean and

Riloff 2004; McCallum and Wellner 2004; Yang, Su, and Tan 2005; Haghighi and Klein 2007)).

Some attempts to automate the template generation process will be discussed in Section 21.4.4.

21.4 Learning-based Approaches to IE

As we discussed in Section 21.3, early information extraction systems used hand-crafted patterns and rules, often encoded in cascaded finite-state transducers. Hand-built IE systems were effective, but manually creating the patterns and rules was extremely time-consuming. For example, it was estimated that it took approximately 1500 person-hours of effort to create the patterns used by the UMass MUC-4 system (Riloff 1993; Lehnert, Cardie, Fisher, McCarthy, Riloff, and Soderland 1992)

Consequently, researchers began to use statistical techniques and machine learning algorithms to automatically create IE systems for new domains. In the following sections, we overview four types of learning-based IE methods: supervised learning of patterns and rules, supervised learning of sequential IE classifiers, weakly supervised and unsupervised learning methods for IE, and learning-based approaches for more global or discourse-oriented approaches to IE.

21.4.1 Supervised Learning of Extraction Patterns and Rules

Supervised learning methods originally promised to dramatically reduce the knowledge engineering bottleneck required to create an IE system for a new domain. Instead of painstakingly writing patterns and rules by hand, knowledge engineering could be reduced to the manual annotation of a collection of training texts. The hope was that a training set could be annotated in a matter of weeks, and nearly anyone with knowledge of the domain could do the annotation work.[6] As we will acknowledge in Section 21.4.3, manual annotation is itself a substantial endeavor, and a goal of recent research efforts is to eliminate this bottleneck as well. But supervised learning methods were an important first step toward automating the creation of information extraction systems.

The earliest pattern learning systems used specialized techniques, sometimes coupled with small amounts of manual effort. AutoSlog (Riloff 1993) and PALKA (Kim and Moldovan 1993) were the first IE pattern learning

[6] In contrast, creating IE patterns and rules by hand typically requires computational linguists who understand how the patterns or rules will be integrated into the NLP system.

systems. AutoSlog (Riloff 1993; Riloff 1996a) matches a small set of syntactic templates against the text surrounding a desired extraction and creates one (or more) lexico-syntactic patterns by instantiating the templates with the corresponding words in the sentence. A "human in the loop" must then manually review the patterns to decide which ones are appropriate for the IE task. PALKA (Kim and Moldovan 1993) uses manually defined frames and keywords that are provided by a user and creates IE patterns by mapping clauses containing the keywords onto the frame's slots. The patterns are generalized based on the semantic features of the words.

Several systems use rule learning algorithms to automatically generate IE patterns from annotated text corpora. LIEP (Huffman 1996) creates candidate patterns by identifying syntactic paths that relate the role fillers in a sentence. The patterns that perform well on training examples are kept, and as learning progresses they are generalized to accommodate new training examples by creating disjunctions of terms. CRYSTAL (Soderland, Fisher, Aseltine, and Lehnert 1995) learns extraction rules using a unification-based covering algorithm. CRYSTAL's rules are "concept node" structures that include lexical, syntactic, and semantic constraints. WHISK (Soderland 1999) was an early system that was specifically designed to be flexible enough to handle structured, semi-structured, and unstructured texts. WHISK learns regular expression rules that consist of words, semantic classes, and wildcards that match any token. $(LP)^2$ (Ciravegna 2001) induces two different kinds of IE rules: *tagging rules* to label instances as desired extractions, and *correction rules* to correct mistakes made by the tagging rules. Freitag created a rule-learning system called SRV (Freitag 1998b) and later combined it with a rote learning mechanism and a Naive Bayes classifier to explore a multi-strategy approach to IE (Freitag 1998a).

Relational learning methods have also been used to learn rule-like structures for IE (e.g., (Roth and Yih 2001; Califf and Mooney 2003; Bunescu and Mooney 2004; Bunescu and Mooney 2007)). RAPIER (Califf and Mooney 1999; Califf and Mooney 2003) uses relational learning methods to generate IE rules, where each rule has a pre-filler, filler, and post-filler component. Each component is a pattern that consists of words, POS tags, and semantic classes. Roth and Yih (Roth and Yih 2001) propose a knowledge representation language for propositional relations and create a 2-stage classifier that first identifies candidate extractions and then selects the best ones. Bunescu and Mooney (Bunescu and Mooney 2004) use Relational Markov Networks to represent dependencies and influences across entities and extractions.

IE pattern learning methods have also been developed for related applications such as question answering (Ravichandran and Hovy 2002), where the goal is to learn patterns for specific types of questions that involve relations between entities (e.g., identifying the birth year of a person).

21.4.2 Supervised Learning of Sequential Classifier Models

An alternative approach views information extraction as a classification problem that can be tackled using sequential learning models. Instead of using explicit patterns or rules to extract information, a machine learning classifier is trained to sequentially scan text from left to right and label each word as an extraction or a non-extraction. A typical labeling scheme is called IOB, where each word is classified as an 'I' if it is inside a desired extraction, 'O' if it is outside a desired extraction, or 'B' if it is the beginning of a desired extraction. The sentence below has been labeled with IOB tags corresponding to phrases that should be extracted as facts about a bombing incident.

> Alleged/B guerrilla/I urban/I commandos/I launched/O two/B highpower/I bombs/I against/O a/B car/I dealership/I in/O downtown/O San/B Salvador/I this/B morning/I.

In the example above, the IOB tags indicate that five phrases should be extracted: "Alleged guerrilla urban commandos", "two highpower bombs", "a car dealership", "San Salvador", and "this morning". Note that the 'B' tag is important to demarcate where one extraction begins and another one ends, particularly in the case when two extractions are adjacent. For example, if only 'I' and 'O' tags were used, then "San Salvador" and "this morning" would run together and appear to be a single extraction. Depending on the learning model, a different classifier may be trained for each type of information to be extracted (e.g., one classifier might be trained to identify perpetrator extractions, and another classifier may be trained to identify location extractions). Or a single classifier can be trained to produce different types of IOB tags for the different kinds of role fillers (e.g., $B_{perpetrator}$ and $B_{location}$) (Chieu and Ng 2002).

A variety of sequential classifier models have been developed using Hidden Markov Models (Freitag and McCallum 2000; Yu, Guan, and Zhou 2005; Gu and Cercone 2006), Maximum Entropy Classifiers (Chieu and Ng 2002), Conditional Random Fields (Peng and McCallum 2004; Choi, Cardie, Riloff, and Patwardhan 2005), and Support Vector Machines (Zelenko, Aone, and Richardella 2003; Finn and Kushmerick 2004; Li, Bontcheva, and Cunningham 2005; Zhao and Grishman 2005). Freitag and McCallum (Freitag and McCallum 2000) use Hidden Markov Models and developed a method to automatically explore different structures for the HMM during the learning process. Gu and Cercone (Gu and Cercone 2006) use HMMs in a 2-step IE process: one HMM retrieves relevant text segments that likely contain a filler, and a second HMM identifies the words to be extracted in these text segments. Finn and Kushmerick (Finn and Kushmerick 2004) also use a 2-step IE process but in a different way: one SVM classifier identifies start and end tags for extractions, and a second SVM looks at tags that were orphaned (i.e., a start tag was found without a corresponding end tag, or vice versa) and tries to identify the missing tag. The second classifier aims to improve IE recall by

producing extractions that otherwise would have been missed. Yu et al. (Yu, Guan, and Zhou 2005) created a cascaded model of HMMs and SVMs. In the first pass, an HMM segments resumes into blocks that represent different types of information. In the second pass, HMMs and SVMs extract information from the blocks, with different classifiers trained to extract different types of information.

The chapter on Fundamental Statistical Techniques in this book explains how to create classifiers and sequential prediction models using supervised learning techniques.

21.4.3 Weakly Supervised and Unsupervised Approaches

Supervised learning techniques substantially reduced the manual effort required to create an IE system for a new domain. However, annotating training texts still requires a substantial investment of time, and annotating documents for information extraction can be deceptively complex (Riloff 1996b). Furthermore, since IE systems are domain-specific, annotated corpora cannot be reused: a new corpus must be annotated for each domain.

To further reduce the knowledge engineering required to create an IE system, several methods have been developed in recent years to learn extraction patterns using weakly supervised and unsupervised techniques. AutoSlog-TS (Riloff 1996b) is a derivative of AutoSlog that requires as input only a preclassified training corpus in which texts are identified as relevant or irrelevant with respect to the domain but are not annotated in any other way. AutoSlog-TS's learning algorithm is a two-step process. In the first step, AutoSlog's syntactic templates are applied to the training corpus exhaustively, which generates a large set of candidate extraction patterns. In the second step, the candidate patterns are ranked based on the strength of their association with the relevant texts. Ex-Disco (Yangarber, Grishman, Tapanainen, and Huttunen 2000) took this approach one step further by eliminating the need for a preclassified text corpus. Ex-Disco uses a small set of manually defined seed patterns to partition a collection of unannotated text into relevant and irrelevant sets. The pattern learning process is then embedded in a bootstrapping loop where (1) patterns are ranked based on the strength of their association with the relevant texts, (2) the best pattern(s) are selected and added to the pattern set, and (3) the corpus is re-partitioned into new relevant and irrelevant sets. Both AutoSlog-TS and Ex-Disco produce IE patterns that performed well in comparison to pattern sets used by previous IE systems. However, the ranked pattern lists produced by these systems still need to be manually reviewed.[7]

Stevenson and Greenwood (Stevenson and Greenwood 2005) also begin with

[7] The human reviewer discards patterns that are not relevant to the IE task and assigns an event role to the patterns that are kept.

seed patterns and use semantic similarity measures to iteratively rank and select new candidate patterns based on their similarity to the seeds. Stevenson and Greenwood use predicate-argument structures as the representation for their IE patterns, as did Surdeanu et al. (Surdeanu, Harabagiu, Williams, and Aarseth 2003) and Yangarber (Yangarber 2003) in earlier work. Sudo et al. (Sudo, Sekine, and Grishman 2003) created an even richer *subtree model* representation for IE patterns, where an IE pattern can be an arbitrary subtree of a dependency tree. The subtree patterns are learned from relevant and irrelevant training documents. Bunescu and Mooney (Bunescu and Mooney 2007) developed a weakly supervised method for relation extraction that uses Multiple Instance Learning (MIL) techniques with SVMs and string kernels.

Meta-bootstrapping (Riloff and Jones 1999) is a bootstrapping method that learns information extraction patterns and also generates noun phrases that belong to a semantic class at the same time. Given a few seed nouns that belong to a targeted semantic class, the meta-bootstrapping algorithm iteratively learns a new extraction pattern and then uses the learned pattern to hypothesize additional nouns that belong to the semantic class. The patterns learned by meta-bootstrapping are more akin to named entity recognition patterns than event role patterns, however, because they identify noun phrases that belong to general semantic classes, irrespective of any events.

Recently, Phillips and Riloff (Phillips and Riloff 2007) showed that bootstrapping methods can be used to learn event role patterns by exploiting *role-identifying nouns* as seeds. A role-identifying noun is a word that, by virtue of its lexical semantics, identifies the role that the noun plays with respect to an event. For example, the definition of the word *kidnapper* is the agent of a kidnapping event. By using role-identifying nouns as seeds, the Basilisk bootstrapping algorithm (Thelen and Riloff 2002) can be used to learn both event extraction patterns as well as additional role-identifying nouns.

Finally, Shinyama and Sekine (Shinyama and Sekine 2006) have developed an approach for completely unsupervised learning of information extraction patterns. Given texts for a new domain, relation discovery methods are used to preemptively learn the types of relations that appear in domain-specific documents. The On-Demand Information Extraction (ODIE) system (Sekine 2006) accepts a user query for a topic, dynamically learns IE patterns for salient relations associated with the topic, and then applies the patterns to fill in a table with extracted information related to the topic.

21.4.4 Discourse-oriented Approaches to IE

Most of the IE systems that we have discussed thus far take a relatively localized approach to information extraction. The IE patterns or classifiers focus only on the local context surrounding a word or phrase when making an extraction decision. Recently, some systems have begun to take a more global view of the extraction process. Gu and Cercone (Gu and Cercone 2006) and Patwardhan & Riloff (Patwardhan and Riloff 2007) use classifiers to first

identify the event-relevant sentences in a document and then apply an IE system to extract information from those relevant sentences.

Finkel et al. (Finkel, Grenager, and Manning 2005) impose penalties in their learning model to enforce label consistency among extractions from different parts of a document. Maslennikov and Chua (Maslennikov and Chua 2007) use dependency and RST-based discourse relations to connect entities in different clauses and find long-distance dependency relations.

Finally, as we discussed in Section 21.3.5, IE systems that process multiple-event documents need to generate multiple templates. Template generation for multiple events is extremely challenging, and only a few learning systems have been developed to automate this process for new domains. WRAP-UP (Soderland and Lehnert 1994) was an early supervised learning system that uses a collection of decision trees to make a series of discourse decisions to automate the template generation process. More recently, Chieu et al. (Chieu, Ng, and Lee 2003) developed a system called ALICE that generates complete templates for the MUC-4 terrorism domain (MUC-4 Proceedings 1992). ALICE uses a set of classifiers that identify extractions for each type of slot and a *template manager* to decide when to create a new template. The template manager uses general-purpose rules (e.g., a conflicting date will spawn a new template) as well as automatically derived "seed words" that are associated with different incident types to distinguish between events.

21.5 How Good is Information Extraction?

Extracting information about events from free text is a challenging problem that is still far from solved. Figure 21.3 illustrates how the various MUC systems progressed from year to year. The vertical axis is precision, and the horizontal axis is recall. We have plotted the top one-third of the system scores in the small ellipse and the top two-thirds in the large ellipse.

We can see that between MUC-3 and MUC-4, the top systems moved up from the high 40s to the high 50s. The principal difference in MUC-5 is that more systems are in the high 50s. By MUC-6 the top two-thirds of the systems are all in a tight cluster with recall in the high 50s and precision in the low 60s. The principal difference between MUC-6 and MUC-7 is that in MUC-7 there were fewer participants.

This is a picture of hill-climbing, where there is a 60% barrier that determines the top of the hill. The tasks in these evaluations were somewhat different, as were the corpora, nevertheless they all seemed to exhibit a ceiling around 60% recall and precision. Although good progress has been made in automating the construction of IE systems using machine learning techniques, current state-of-the-art systems still have not broken through this 60% barrier

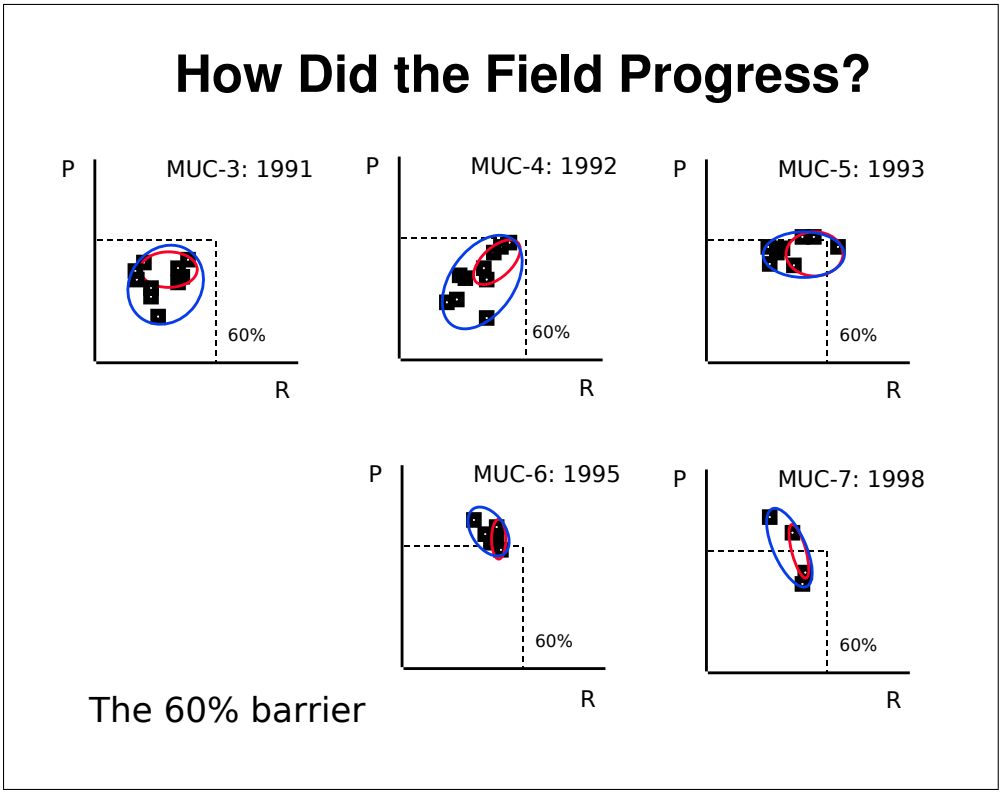

FIGURE 21.3: Chronology of MUC system performance

in performance on the MUC data sets (e.g., (Soderland 1999; Chieu, Ng, and Lee 2003; Maslennikov and Chua 2007)).[8]

There are several possible explanations for this barrier. Detailed analysis of the performance of some of the systems revealed that the biggest source of mistakes was in entity and event coreference; more work certainly needs to be done on this. Another possibility is that 60% is what the text wears on its sleeve; the rest is implicit and requires inference and access to world knowledge.

Another explanation is that there is a Zipf distribution of problems that need to be solved. When we solve the more common problems, we get a big boost in performance. But we have solved all the most common problems, and now we are in the long tail of the distribution. We might take care of

[8]The one exception is that (Maslennikov and Chua 2007) report an F score of 72% on a modified version of the MUC-6 corpus.

a dozen new problems we find in the training data, only to find that none of these problems occur in the test data, so there is no effect on measured performance. One possible solution is active learning (e.g., (Lewis and Catlett 1994; Liere and Tadepalli 1997; McCallum and Nigam 1998; Thompson, Califf, and Mooney 1999)) and the automated selection of rare training examples in the tail for additional manual annotation. This could help to reduce the overall amount of annotated training data that is required, while still adequately covering the rare cases.

A final possibility is both simple and disconcerting. Good named entity recognition systems typically recognize about 90% of the entities of interest in a text, and this is near human performance. To recognize an event and its arguments requires recognizing about four entities, and $.90^4$ is about 60%. If this is the reason for the 60% barrier, it is not clear what we can do to overcome it, short of solving the general natural language problem in a way that exploits the implicit relations among the elements of a text.

21.6 Acknowledgments

This work was supported in part by Department of Homeland Security Grant N0014-07-1-0152. We are grateful to Doug Appelt, Ray Mooney, and Siddharth Patwardhan, who provided extremely helpful comments on an earlier draft of this chapter.

Bibliography

Ananiadou, S., C. Friedman, and J. Tsujii (2004). Introduction: Named Entity Recognition in Biomedicine. *Journal of Biomedical Informatics 37*(6).

Ananiadou, S. and J. McNaught (Eds.) (2006). *Text Mining for Biology and Biomedicine*. Artech House, Inc.

Aone, C. and S. W. Bennett (1996). Applying machine learning to anaphora resolution. In S. Wermter, E. Riloff, and G. Scheler (Eds.), *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, pp. 302–314. Springer-Verlag, Berlin.

Bagga, A. and B. Baldwin (1998). Entity-based Cross-Document Coreferencing using the Vector Space Model. In *Proceedings of the 17th International Conference on Computational Linguistics*.

Banko, M., M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni (2007). Open Information Extraction from the Web.

Bean, D. and E. Riloff (2004). Unsupervised Learning of Contextual Role Knowledge for Coreference Resolution. In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL 2004)*.

Bikel, D. M., R. Schwartz, and R. M. Weischedel (1999). An Algorithm that Learns What's in a Name. *Machine Learning 34*.

Brin, S. (1998). Extracting Patterns and Relations from the World Wide Web. In *WebDB Workshop at EDBT-98*.

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

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

Califf, M. and R. Mooney (1999). Relational Learning of Pattern-matching Rules for Information Extraction. In *Proceedings of the 16th National Conference on Artificial Intelligence*.

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

Cardie, C. and K. Wagstaff (1999). Noun Phrase Coreference as Clustering. In *Proc. of the Joint Conference on Empirical Methods in NLP and Very Large Corpora*.

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

Chieu, H., H. Ng, and Y. Lee (2003). Closing the Gap: Learning-Based Information Extraction Rivaling Knowledge-Engineering Methods. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics*.

Choi, Y., C. Cardie, E. Riloff, and S. Patwardhan (2005). 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*, pp. 355–362.

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

Collins, M. and Y. Singer (1999). Unsupervised Models for Named Entity Classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*.

Croft, W. A. (1991). *Syntactic Categories and Grammatical Relations*. Chicago, Illinois: University of Chicago Press.

Cucerzan, S. and D. Yarowsky (1999). Language Independent Named Entity Recognition Combining Morphologi cal and Contextual Evidence. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*.

Cunningham, H., D. Maynard, K. Bontcheva, and V. Tablan (2002). GATE: A framework and graphical development environment for robust nlp tools and applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Dagan, I. and A. Itai (1990). Automatic Processing of Large Corpora for the Resolution of Anaphora References. In *Proceedings of the Thirteenth International Conference on Computational Linguistics (COLING-90)*, pp. 330–332.

Etzioni, O., M. Cafarella, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates (2005). Unsupervised Named-Entity Extraction from the Web: An Experimental Study. *Artificial Intelligence 165*(1), 91–134.

Finkel, J., T. Grenager, and C. Manning (2005, June). 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, pp. 363–370.

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

Fleischman, M. and E. Hovy (2002, August). Fine grained classification of named entities. In *Proceedings of the COLING conference*.

Fleischman, M., E. Hovy, and A. Echihabi (2003). Offline strategies for online question answering: Answering questions before they are asked. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics*.

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

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

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

Friedman, C. (1986). *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*, Chapter Automatic Structuring of Sublanguage Information. Lawrence Erlbaum Associates.

Gooi, C. and J. Allan (2004). Cross-Document Coreference on a Large Scale Corpus. In *Proceedings of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL 2004)*.

Grishman, R., S. Huttunen, and R. Yangarber (2002). Real-Time Event Extraction for Infectious Disease Outbreaks. In *Proceedings of HLT 2002 (Human Language Technology Conference)*.

Gu, Z. and N. Cercone (2006, July). 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, pp. 481–488.

Haghighi, A. and D. Klein (2007). Unsupervised Coreference Resolution in a Nonparametric Bayesian Model. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

Harabagiu, S., R. Bunescu, and S. Maiorana (2001). Text and Knowledge Mining for Coreference Resolution. In *Proceedings of the The Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Hirschman, L., A. Yeh, C. Blaschke, and A. Valencia (2005, May). Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics 6(Suppl 1)*.

Hobbs, J. R., D. E. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson (1997). FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. In E. Roche and Y. Schabes (Ed.), *Finite State Devices for Natural Language Processing*, pp. 383–406. MIT Press.

Hobbs, J. R., D. E. Appelt, J. Bear, D. Israel, and M. Tyson (1992). FASTUS: A System for Extracting Information from Natural-Language Text. SRI Technical Note 519, SRI International, Menlo Park, California.

Huffman, S. (1996). Learning Information Extraction Patterns from Examples. In S. Wermter, E. Riloff, and G. Scheler (Eds.), *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, pp. 246–260. Springer-Verlag, Berlin.

Igo, S. and E. Riloff (2008). Learning to Identify Reduced Passive Verb Phrases with a Shallow Parser. In *Proceedings of the 23rd National Conference on Artificial Intelligence*.

Joshi, A. K. (1996). A Parser from Antiquity: An Early Application of Finite State Transducers to Natural Language Parsing. In *European Conference on Artificial Intelligence 96 Workshop on Extended Finite State Models of Language*, pp. 33–34.

Kehler, A. (1997). Probabilistic Coreference in Information Extraction. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*.

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

Lehnert, W., C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland (1992). University of Massachusetts: Description of the CIRCUS System as Used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, San Mateo, CA, pp. 282–288. Morgan Kaufmann.

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

Lewis, D. D. and J. Catlett (1994). Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the Eleventh International Conference on Machine Learning*.

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

Liere, R. and P. Tadepalli (1997). Active learning with committees for text categorization. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*.

Light, M., G. Mann, E. Riloff, and E. Breck (2001). Analyses for Elucidating Current Question Answering Technology. *Journal for Natural Language Engineering 7*(4).

Mann, G. and D. Yarowsky (2003). Unsupervised Personal Name Disambiguation. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003)*.

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

Mayfield, J., D. Alexander, B. Dorr, J. Eisner, T. Elsayed, T. Finin, C. Fink, M. Freedman, N. Garera, P. McNamee, S. Mohammad, D. Oard, C. Piatko, A. Sayeed, Z. Syed, R. Weischedel, T. Xu, and D. Yarowsky (2009). Cross-Document Coreference Resolution: A Key Technology for Learning by Reading. In *Working Notes of the AAAI 2009 Spring Symposium on Learning by Reading and Learning to Read*.

McCallum, A. and B. Wellner (2004). Conditional Models of Identity Uncertainty with Application to Noun Coreference. In *18th Annual Conference on Neural Information Processing Systems*.

McCallum, A. K. and K. Nigam (1998). Employing EM and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*.

McCarthy, J. and W. Lehnert (1995). Using Decision Trees for Coreference Resolution. In *Proc. of the Fourteenth International Joint Conference on Artificial Intelligence*.

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

MUC-5 Proceedings (1993). *Proceedings of the Fifth Message Understanding Conference (MUC-5)*. San Francisco, CA.

MUC-6 Proceedings (1995). *Proceedings of the Sixth Message Understanding Conference (MUC-6)*.

MUC-7 Proceedings (1998). *Proceedings of the Seventh Message Understanding Conference (MUC-7)*.

Ng, V. and C. Cardie (2002). Improving Machine Learning Approaches to Coreference Resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Niu, C., W. Li, and R. K. Srihari (2004). Weakly Supervised Learning for Cross-Document Person Name Disambiguation Supported by Information Extraction. In *Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics*.

Pasca, M. (2007). Weakly-supervised Discovery of Named Entities using Web Search Queries. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM-07), Lisboa, Portugal*, pp. 683–690.

Pasca, M., D. Lin, J. Bigham, A. Lifchits, and A. Jain (2006). Names and Similarities on the Web: Fact Extraction in the Fast Lane. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL-06), Sydney, Australia*.

Patwardhan, S. and E. Riloff (2007). Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions. In *Proceedings of 2007*

the Conference on Empirical Methods in Natural Language Processing (EMNLP-2007).

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

Phillips, W. and E. Riloff (2007). Exploiting Role-Identifying Nouns and Expressions for Information Extraction. In *Proceedings of the 2007 International Conference on Recent Advances in Natural Language Processing (RANLP-07)*, pp. 468–473.

Ravichandran, D. and E. Hovy (2002). Learning Surface Text Patterns for a Question Answering System. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*.

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

Riloff, E. (1996a). An Empirical Study of Automated Dictionary Construction for Information Extraction in Three Domains. *Artificial Intelligence 85*, 101–134.

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

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

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

Sang, E. F. T. K. and F. D. Meulder (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pp. 142–147.

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

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

Soderland, S. (1999). Learning Information Extraction Rules for Semi-structured and Free Text. *Machine Learning*.

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

Soderland, S. and W. Lehnert (1994). Wrap-Up: A trainable discourse module for information extraction. *Journal of Artificial Intelligence Research (JAIR) 2*, 131–158.

Soon, W., H. Ng, and D. Lim (2001). A Machine Learning Approach to Coreference of Noun Phrases. *Computational Linguistics 27*(4), 521–541.

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

Strassel, S., M. Przybocki, K. Peterson, Z. Song, and K. Maeda (2008). Linguistic Resources and Evaluation Techniques for Evaluation of Cross-Document Automatic Content Extraction. In *Proceedings of the Sixth International Language Resources and Evaluation Conference (LREC-08)*.

Subramaniam, L. V., S. Mukherjea, P. Kankar, B. Srivastava, V. S. Batra, P. V. Kamesam, and R. Kothari (2003). Information extraction from biomedical literature: Methodology, evaluation and an application. In *CIKM '03: Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pp. 410–417.

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

Surdeanu, M., S. Harabagiu, J. Williams, and P. Aarseth (2003). Using predicate-argument structures for information extraction. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics*.

Thelen, M. and E. Riloff (2002). A Bootstrapping Method for Learning Semantic Lexicons Using Extraction Pa ttern Contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pp. 214–221.

Thompson, C. A., M. E. Califf, and R. J. Mooney (1999). Active learning for natural language parsing and information extraction. In *Proceedings of the Sixteenth International Conference on Machine Learning*.

Yakushiji, A., Y. Miyao, T. Ohta, and J. Tateisi, Y. Tsujii (2006). Automatic construction of predicate-argument structure patterns for biomedical information extraction. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*.

Yang, X., J. Su, and C. L. Tan (2005). Improving Pronoun Resolution using Statistics-Based Semantic Compatibility Information. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics*.

Yangarber, R. (2003). Counter-training in the discovery of semantic patterns. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics*.

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

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

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

Zhao, S. and R. Grishman (2005). Extracting Relations with Integrated Information Using Kernel Methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05), Ann Arbor, Michigan*.