In 2007 Proceedings of Recent Advances in Natural Language Processing (RANLP-07)

Exploiting Role-Identifying Nouns and Expressions for Information Extraction

William Phillips and Ellen Riloff
School of Computing
University of Utah
Salt Lake City, UT 84112
{*phillips,riloff*}@*cs.utah.edu*

Abstract

We present a new approach for extraction pattern learning that exploits *role-identifying nouns*, which are nouns whose semantics reveal the role that they play in an event (e.g., an "assassin" is a perpetrator). Given a few seed nouns, a bootstrapping algorithm automatically learns role-identifying nouns, which are then used to learn extraction patterns. We also introduce a method to learn *role-identifying expressions*, which consist of a role-identifying verb linked to an event (e.g., *"<subject> participated in the murder"*). We present experimental results on the MUC-4 terrorism corpus and a disease outbreaks corpus.

Keywords

information extraction, learning, event roles

1 Introduction

Our research focuses on event-based information extraction, where the task is to identify facts related to events. Event-based information extraction systems have been developed for many domains, including terrorism [8, 3, 10, 13], management succession [17], corporate acquisitions [5, 6], and disease outbreaks [7]. Many IE systems rely on extraction patterns or rules, such as CRYSTAL [13], AutoSlog/AutoSlog-TS [9, 10], RAPIER [2], WHISK [12], Ex-DISCO [17], Snowball [1], $(LP)^2$ [4], Subtree patterns [14], and predicate-argument rules [16].

Our work presents a new approach for IE pattern learning that takes advantage of *role-identifying nouns*, *role-identifying verbs*, and *role-identifying expressions*. We will refer to a word or phrase as being *role-identifying* if it reveals the role that an entity or object plays in an event. For example, the word *assassin* is a role-identifying noun because an assassin is the perpetrator of an event, by definition. Similarly, the verb *participated* is a role-identifying verb because it means that someone played the role of actor (agent) in an activity. When a role-identifying verb is explicitly linked to an event noun, we have a role-identifying expression. For example, *"<subject> participated in the murder"* means that the subject of "participated" is a perpetrator of the murder event.

We have developed a new approach to IE pattern learning that exploits role-identifying nouns. We employ the Basilisk bootstrapping algorithm [15] to learn role-identifying nouns, and then use them to rank extraction patterns. We also describe a learning process that creates a new type of extraction pattern that captures role-identifying expressions. This process begins by automatically inducing event nouns from a corpus via bootstrapping. We then generate patterns that extract an event noun as a syntactic argument. Finally, we match these event patterns against a corpus and generate expanded patterns for each syntactic dependency that is linked to the pattern's verb.

This paper is organized as follows. Section 2 gives the motivation for role-identifying nouns and expressions. Section 3 describes the extraction pattern learning process. Section 4 presents our experimental results, and Section 5 discusses related work.

2 Motivation

Our work is motivated by the idea that *role-identifying nouns* and *role-identifying expressions* can be beneficial for information extraction. In this section, we explain what they are and how we aim to use them.

2.1 Role-Identifying Nouns

Our research exploits nouns that, by definition, identify the role that the noun plays with respect to an event. For example, the word *kidnapper* is defined as the perpetrator of a kidnapping. Similarly, the word *victim* is defined as the object of a violent event. We will refer to these nouns as **Lexically Role-Identifying Nouns** because their lexical meaning identifies the role that the noun plays in some event.

We have observed that there are a surprisingly large number of role-identifying nouns. For example, the words *arsonist*, *assassin*, *kidnapper*, *robber*, and *sniper* refer to perpetrators of a crime. Similarly, the words *casualty*, *fatality*, *victim*, and *target* refer to objects of a violent event. It is important to note that in a sentence these nouns may serve in a different thematic role associated with a verb. For example, in *"The assassin was arrested"*, the assassin is the theme of the verb "arrest", but it is also understood to be the perpetrator of an (implicit) assassination event. Our work focuses on high-level **event roles**, rather than thematic (semantic) roles that represent verb arguments.

Within a specific domain, some words can also be inferred to serve in an event role based on their general semantic class. For example, consider disease outbreak reports. If a toddler is mentioned, one can reasonably infer that the toddler is a victim of a disease outbreak. The reason is that toddlers cannot fill any other roles commonly associated with disease outbreaks (e.g., they cannot be medical practitioners, scientists, or spokespeople). The intuition comes from Grice's Maxim of Relevance: any reference to a child in a disease report is almost certainly a reference to a victim because the child wouldn't be relevant to the story otherwise. As another example, if a restaurant is mentioned in a crime report, then a crime probably occurred in or around the restaurant. Of course, context can always provide another explanation (e.g., the restaurant could be the place where a suspect was arrested). But generally speaking, if a word's semantics are compatible with only one role associated with an event, then we often infer that it is serving in that role. We will refer to nouns that strongly evoke one event role as **Semantically Role-Identifying Nouns**.

Role-identifying nouns are often not the most desirable extractions for an IE system because they are frequently referential. For example, "the assassin" may be coreferent with a proper name (e.g., "Lee Harvey Oswald"), which is a more desirable extraction. However, role-identifying nouns can be exploited for extraction pattern learning. Our intuition is that if a pattern consistently extracts role-identifying nouns associated with one event role, then the pattern is probably a good extractor for that role.

2.2 Role-Identifying Expressions

For event-based information extraction, the most reliable IE patterns usually depend on a word that explicitly refers to an event. For example, the pattern *"<subject> was kidnapped"* indicates that a kidnapping took place, and the subject of "kidnapped" is extracted as the victim. In contrast, some verbs identify a role player associated with an event without referring to the event itself. For example, consider the verb "participated". By its definition, "participated" means that someone took part in something, so the pattern *"<subject> participated"* identifies the actor (agent) of an activity. However, the word "participate" does not reveal what the activity is. The activity is often specified in another argument of the verb (e.g., *"John participated in the debate."*). In other cases, the event must be inferred through discourse (e.g., *"The debate took place at Dartmouth. John participated."*).

Our observation is that there are many verbs whose main purpose is to identify a role player associated with an event, without defining the event itself. We will refer to them as **Role-Identifying Verbs**. Some additional examples of role-identifying verbs are "perpetrated", "accused", and "implicated", which all identify the (alleged) perpetrator of an event. Often, the agent of the verb is also the agent of the (implicit) event. For example, the agents of "participated" and "perpetrated" are also the agents of the event (e.g., *"John perpetrated the attack"*). However, an entity or object can function in one thematic role with respect to the verb and a different role with respect to the event. For example, in the sentence *"John was implicated in the attack"*, the theme of "implicated" is the (alleged) agent of the attack.

Our goal is to use role-identifying verbs in extraction patterns. The challenge is that these verbs are generally not reliable extractors by themselves because it is crucial to know what event they are referring to. For example, *"John participated in the bombing"* is relevant to a terrorism IE task, but *"John participated in the meeting"* is not. Our solution is to create patterns that include both a role-identifying verb and a relevant event noun as a syntactic argument to the verb. We will refer to these patterns as **Role-Identifying Expression (RIE) patterns**.

3 Extraction Pattern Learning

3.1 Overview

Our hypothesis is that role-identifying nouns can be valuable for extraction pattern learning. Throughout this work, we rely heavily on the Basilisk bootstrapping algorithm [15], which was originally designed for semantic lexicon induction (i.e., to learn which nouns belong to a general semantic category, such as ANIMAL or VEHICLE). In Section 3.2.2, we will use Basilisk as it was originally intended – to generate nouns belonging to the semantic category EVENT. However, we also use Basilisk in a new way – to learn role-identifying nouns.

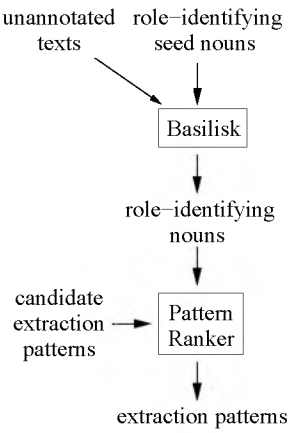

Fig. 1: *The Extraction Pattern Learning Process*

Fig. 1 shows the high-level process for extraction pattern learning. First, we use Basilisk to generate role-identifying nouns for an event role associated with the IE task. Next, we create a large set of *candidate patterns* by exhaustively generating all extraction patterns that occur in the training corpus. Finally, we rank the candidate patterns based on their tendency to extract the role-identifying nouns. This learning process is therefore very weakly supervised: only an unannotated corpus and a small set of role-identifying seed nouns are needed to learn extraction patterns for an event role.

In the following sections, we explain how two types of candidate patterns are generated, how Basilisk learns role-identifying nouns, and how the role-identifying nouns are used to select the best patterns.

3.2 Generating Candidate Patterns

Our goal is to learn two different kinds of extraction patterns. First, we generate the traditional kind of patterns which extract information from the arguments of verbs and nouns that describe an event (e.g., *"<subject> was kidnapped"* or *"assassination of <np>"*). Second, we generate a new type of extraction pattern that captures *role-identifying expressions*.

3.2.1 Generating Standard Patterns

We use the AutoSlog extraction pattern learner [9] to generate candidate "traditional" extraction patterns. AutoSlog applies syntactic heuristics to automatically learn lexico-syntactic patterns from annotated noun phrases. For example, consider the sentence *"A turkey in Indonesia was recently infected with avian flu."* If *"A turkey"* is labeled as a disease victim, then AutoSlog will create the pattern *"<subject> PassVP(infected)"* to extract victims. This pattern matches instances of the verb "infected" in the passive voice, and extracts the verb's subject as a victim.

We use AutoSlog in an unsupervised fashion by applying it to unannotated texts and generating a pattern to extract (literally) every noun phrase in the corpus. We will refer to the resulting set of patterns as the *candidate standard IE patterns*.

3.2.2 Generating RIE Patterns

Fig. 2 shows the process for generating candidate Role-Identifying Expression (RIE) patterns, which involves two steps. In Step 1, we use the Basilisk semantic lexicon learner [15] to generate *event nouns*, which are nouns that belong to the semantic category EVENT (e.g., "assassination"). This step may not be needed if a list of event nouns for the domain is already available or can be obtained from a resource such as WordNet. However, we use Basilisk to demonstrate that event nouns for a domain can be automatically generated. As input, Basilisk requires just a few seed nouns and an unannotated text corpus. We explain how the seed nouns were chosen in Section 4.1.

We ran Basilisk for 50 iterations, generating 5 event nouns per iteration. However, we are only interested in events that are relevant to the IE task. For example, for the terrorism domain we want to extract information about murder and kidnapping events, but not meetings or celebratory events. So we manually reviewed the event nouns and retained only those that are *relevant* to the IE task. Of the 250 event nouns generated for each domain, we kept 94 for terrorism and 220 for disease outbreaks.[1]

In Step 2, we create the role-identifying expression patterns. Each RIE pattern must be anchored by a verb phrase that has a syntactic argument that is an event noun. We begin by creating standard patterns that can extract events. We give the relevant event nouns to the AutoSlog pattern learner [9] as input,[2] which then creates patterns that can extract

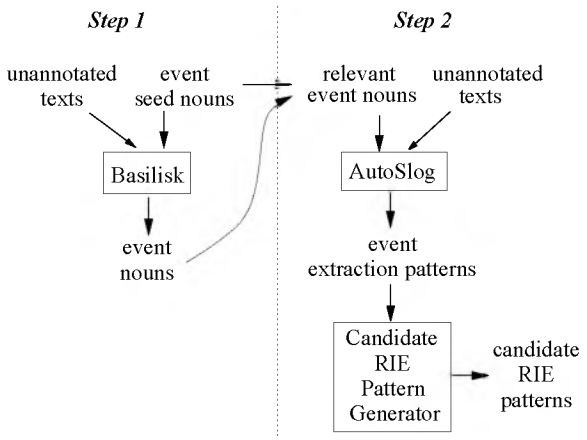

Fig. 2: *Generating Candidate RIE Patterns*

events. The *Candidate RIE Pattern Generator* then expands these event patterns into role-identifying expressions. For each instance of a verb pattern[3], the verb's subject, direct object, and all attached prepositional phrases are identified. For each one, an expanded pattern is spawned that includes this syntactic relation. For example, consider the event pattern *"committed <EVENT>"*, which matches active voice verb instances of "committed" and extracts its direct object as an event (e.g., *"committed the murder"*). Now suppose that this pattern is applied to the sentence: *"John Smith committed the murder in November."* Two syntactic relations are associated with the verb phrase: its subject (*"John Smith"*) and a PP (*"in November"*). The following two candidate RIE patterns would then be generated: *"<subject> committed <EVENT>"* and *"committed <EVENT> in <np>"*.

3.3 Learning Role-Identifying Nouns

Now that we have a large set of candidate extraction patterns, we return to the high-level learning process depicted in Fig. 1. The first step is to generate role-identifying nouns for each event role associated with the IE task. We use the Basilisk bootstrapping algorithm [15], which was originally designed for semantic lexicon induction but its algorithm relies heavily on lexico-syntactic pattern matching, which also seemed well-suited for learning role-identifying nouns.

Basilisk begins with a small set of seed nouns and then iteratively induces more nouns. Each bootstrapping cycle consists of 3 steps: (1) collect a pool of patterns that tend to extract the seeds, (2) collect all nouns extracted by these patterns, (3) score each noun based on the scores of all patterns that extracted it.[4] We tried two different ways of selecting role-identifying seed nouns to kickstart the bootstrapping, which we will discuss in Section 4.1. Below are some of the role-identifying nouns that were learned for terrorism perpetrators and disease outbreak victims:

Terrorism Perpetrator: assailants, attackers, cell, culprits, extremists, hitmen, kidnappers,

[1] Diseases were often used to refer to outbreaks, so we included disease names as event nouns in this domain.

[2] Since AutoSlog is a supervised learner, the event nouns are essentially used to automatically annotate the corpus.

[3] AutoSlog's noun patterns are not used.

[4] We made one minor change to Basilisk's RlogF scoring function, by adding 1 inside the logarithm so that words with frequency 1 would not get a zero score.

Terror PerpInd	*Terror PerpOrg*	*Terror Target*	*Terror Victim*
<subj> riding	<subj> claimed responsibility	destroyed <dobj>	murder of <np>
was kidnapped by <np>	<subj> is group	burned <dobj>	<subj> was killed
was killed by <np>	<subj> claimed	<subj> was damaged	assassination of <np>
<subj> identified themselves	delegates of <np>	awakened with <np>	killed <dobj>
was perpetrated by <np>	was attributed to <np>	blew up <dobj>	<subj> was sacrificed
Outbreak Victim	*Outbreak Disease*	*Terror Weapon*	
brains of <np>	outbreaks of <np>	threw <dobj>	
mother of <np>	woman was diagnosed with <np>	hurled <dobj>	
disease was transmitted to <np>	to contracted <dobj>	confiscated <dobj>	
<subj> is unwell	<subj> hits	rocket <dobj>	
<subj> tests positive	to contract <dobj>	sticks of <np>	

Table 1: *Top 5 Standard Patterns for Each Event Role*

Terror PerpInd	*Terror PerpOrg*	*Terror Target*	*Terror Victim*
EV was perpetrated by <np>	<subj> carried out EV	EV destroyed <dobj>	<subj> was killed in EV
<subj> committed EV	EV was perpetrated by <np>	caused EV to <np>	EV including <dobj>
<subj> was involved in EV	<subj> called for EV	EV damaged <dobj>	<subj> was killed during EV
<subj> participated in EV	EV was attributed to <np>	staged EV on <np>	EV led <dobj>
<subj> involved in EV	EV was carried out by <np>	EV caused to <np>	identified <dobj> after EV
Outbreak Victim	*Outbreak Disease*	*Terror Weapon*	
<subj> was suffering from EV	EV known as <np>	confiscated <dobj> during EV	
<subj> contracted EV	EV called <dobj>	EV was caused by <np>	
EV was transmitted from <dobj>	EV was known as <np>	EV carried out with <np>	
EV infect <dobj>	EV due to <np>	<subj> was thrown by EV	
EV killed dozens of <np>	<subj> was caused by EV	<subj> caused EV	

Table 2: *Top 5 RIE Patterns for Each Event Role (EV = Event Noun)*

militiamen, MRTA, narco-terrorists, sniper

Outbreak Victim: bovines, crow, dead, eagles, fatality, pigs, swine, teenagers, toddlers, victims

Most of the perpetrator words are lexically role-identifying nouns, while most of the disease outbreak victim words are semantically role-identifying nouns.

3.4 Selecting Extraction Patterns

When Basilisk's bootstrapping is done, we have a large collection of role-identifying nouns. Next, we rank all of the candidate extraction patterns based on the same RlogF metric that Basilisk uses internally, which is: $RlogF(p_i) = \frac{f_i}{n_i} * \log_2(f_i)$, where f_i is the number of unique role-identifying nouns extracted by pattern p_i and n_i is the total number of unique nouns extracted by p_i. The top N highest-ranking patterns are selected as the best extractors for the event role.

We used this approach to learn extraction patterns for seven event roles: five roles associated with terrorism (*individual perpetrators*, *organizational perpetrators*, *victims*, *physical targets*, and *weapons*) and two roles associated with disease outbreaks (*diseases* and *victims*). Tables 1 and 2 show the top 5 standard and RIE extraction patterns learned for each event role.

4 Evaluation

We evaluated our performance on two data sets: the MUC-4 terrorist events corpus [8], and a ProMed disease outbreaks corpus. The MUC-4 corpus contains 1700 stories and answer key templates for each story. We focused on five MUC-4 string slots: *perpetrator individuals*, *perpetrator organizations*, *physical targets*, *victims*, and *weapons*. We used 1400 stories for training (DEV+TST1), 100 stories for tuning (TST2), and 200 stories as a blind test set (TST3+TST4).

ProMed-mail[5] is an open-source, global electronic reporting system for outbreaks of infectious diseases. Our ProMed IE data set includes a training set of 4659 articles, and a test set of 120 different articles coupled with answer key templates that we manually created. We focused on extracting *diseases* and *victims*, which can be people, animals, or plants.

The complete IE task involves the creation of answer key templates, one template per incident.[6] 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 pattern learning, so we evaluated the extractions themselves, before template generation would take place. This approach directly measures how accurately the patterns find relevant information, without confounding factors introduced by the template generation process.[7] We used a *head noun* scoring scheme, where an extraction is correct if its head noun matches the head noun in the answer key.[8]

4.1 Seed Word Selection

To select event seed nouns, we shallowly parsed the corpus, sorted the head nouns of NPs based on frequency, and then manually identified the first 10 nouns that represent an event.

To select role-identifying seed nouns, we experimented with two approaches. First, we collected all of the head nouns of NPs in the corpus and sorted them

[5] See www.promedmail.org

[6] Many MUC-4 and ProMed stories mention multiple incidents.

[7] For example, if the coreference resolver incorrectly decides that two items are coreferent and merges them, then it will appear that only one item was extracted by the patterns when in fact both were extracted.

[8] This approach allows for different modifiers in an NP as long as the heads match. We also discarded pronouns because we do not perform coreference resolution.

System	*PerpInd*			*PerpOrg*			*Target*			*Victim*			*Weapon*		
	Rec	Pr	F	Rec	Pr	F	Rec	Pr	F	Rec	Pr	F	Rec	Pr	F
ASlogTS	.49	.35	.41	.33	.49	.40	.64	.42	.51	.52	.48	.50	.45	.39	.42
Top20	.18	.55	.27	.15	.67	.25	.46	.51	**.48**	.30	.51	.38	.40	.59	.47
Top50	.22	.48	.30	.17	.50	.25	.51	.44	.47	.35	.42	.38	.52	.48	**.50**
Top100	.36	.45	**.40**	.21	.52	.30	.59	.37	.46	.42	.37	.39	.53	.43	.48
Top200	.40	.35	.37	.34	.45	**.39**	.64	.29	.40	.48	.35	**.40**	.53	.35	.42

Table 3: *MUC-4 Results for Standard Patterns*

by frequency. For each event role, we then manually identified the first 10 nouns that were role-identifying nouns for that role. We will refer to these as the *high-frequency seeds*.

We also tried using seed patterns instead of seed nouns. For each event role, we manually defined 10 patterns that reliably extract NPs for that role. For example, the pattern *"<subject> kidnapped"* was a seed pattern to identify perpetrators. We also defined an *Other* role to capture other possible roles, using 60 seed patterns for this category in terrorism and 30 for disease outbreaks.[9] We then applied the patterns to the corpus and collected their extractions. For each event role ($erole_i$) and each head noun of an extraction (n), we computed the following probability:

$$Pr(erole_i \mid n) = \frac{|n \text{ extracted by an } erole_i \text{ pattern}|}{\sum_{k=1}^{|E|} |n \text{ extracted by an } erole_k \text{ pattern}|} \quad (1)$$

where E is the number of event roles. All nouns with probability > 0.50 and frequency ≥ 2 were used as seeds. We will refer to these as the *pattern-generated seeds*. The advantages of this approach are that it is natural to think of seed patterns for a role, and a few patterns can yield a large set of seed nouns. The drawbacks are that these nouns may not be frequent words and they are not guaranteed to be role-specific.

Both approaches worked reasonably well, but combining the two approaches worked even better. So for all of our experiments, the seeds consist of the *high-frequency seeds* plus the *pattern-generated seeds*.

4.2 Experimental Results

To establish a baseline for comparison, we trained the AutoSlog-TS IE pattern learner [10] on our two data sets. AutoSlog-TS generates a ranked list of extraction patterns, which needs to be manually reviewed. [10] The first row of Tables 3 and 4 shows its recall, precision, and F-measure. The MUC-4 results are similar to those of ALICE and the other MUC-4 systems as reported in [3], although those results are with template generation so not exactly comparable to ours.

Next, we evaluated the standard IE patterns produced by our learning process. Tables 3 and 4 show the scores obtained for the top 20, 50, 100, and 200 patterns in the ranked list. As one would expect, the first 20 patterns yielded the highest precision. As more patterns are used, recall increases but precision drops. In most cases, the best F-measure scores were achieved with the top 100 or 200 patterns.

[9] We roughly wanted to balance the number of patterns for this role with all of the other roles combined.

[10] We reviewed patterns with score $\geq .951$ and frequency ≥ 3 for terrorism, and score ≥ 5.931 for disease outbreaks.

System	*Disease*			*Victim*		
	Rec	Pr	F	Rec	Pr	F
ASlogTS	.51	.27	.36	.48	.36	.41
Top20	.40	.33	.36	.34	.38	.36
Top50	.44	.33	.38	.35	.38	.36
Top100	.47	.31	.37	.36	.37	**.37**
Top200	.54	.30	**.39**	.38	.33	.35

Table 4: *ProMed Results for Standard Patterns*

We then included the RIE patterns produced by our learning process. First, we combined the top 20 Standard patterns with the RIE patterns. Our expectation was that this set of patterns should have good precision but perhaps only moderate recall. Second, we combined the top 100 Standard patterns with the RIE patterns. We expected this set of patterns to have higher recall but lower precision. In the terrorism domain, fewer than 100 RIE patterns were learned for each event role, so we used them all. For disease outbreaks, many RIE patterns were learned so we evaluated the top 100 and the top 200.

System	*Disease*			*Victim*		
	Rec	Pr	F	Rec	Pr	F
Top20	.40	.33	.36	.34	.38	.36
Top20+100RIEs	.44	.32	.37	.36	.35	.36
Top20+200RIEs	.45	.31	.36	.40	.36	**.38**
Top100	.47	.31	.37	.36	.37	.37
Top100+100RIEs	.50	.31	**.38**	.38	.35	.36
Top100+200RIEs	.50	.30	.37	.41	.35	**.38**
ASlogTS	.51	.27	.36	.48	.36	.41

Table 5: *Promed Results for All Patterns*

Tables 5 and 6 show the results. The RIE patterns were most beneficial for the terrorism perpetrator roles, increasing the F score by +6 for *PerpInd* and +11 for *PerpOrg* when using 20 Standard patterns. The F score also increased by 1-2 points for the terrorism *Victim* and *Weapon* roles, but performance decreased on the *Target* role. For disease outbreaks, the RIE patterns improved the F score for both the *Disease* and *Victim* roles.

The last row of Tables 5 and 6 show the AutoSlog-TS baseline again for comparison. Our IE system is competitive with AutoSlog-TS, which required manual review of its patterns. In contrast, our IE patterns were learned automatically using only seed words and unannotated texts for training.

4.3 Analysis

Table 7 shows examples of RIE patterns that behaved differently from their Standard pattern counterparts. The *Pr* column shows $Pr(erole \mid p)$ for each pattern p, which is the percentage of the pattern's extractions

System	*PerpInd*			*PerpOrg*			*Target*			*Victim*			*Weapon*		
	Rec	Pr	F	Rec	Pr	F	Rec	Pr	F	Rec	Pr	F	Rec	Pr	F
Top20	.18	.55	.27	.15	.67	.25	.46	.51	.48	.30	.51	.38	.40	.59	.47
Top20+RIEs	.25	.48	.33	.25	.70	.36	.46	.42	.44	.32	.48	.38	.41	.60	**.49**
Top100	.36	.45	.40	.21	.52	.30	.59	.37	**.46**	.42	.37	.39	.53	.43	.48
Top100+RIEs	.40	.43	**.41**	.30	.57	**.40**	.59	.33	.42	.44	.36	**.40**	.53	.43	.48
ASlogTS	.49	.35	.41	.33	.49	.40	.64	.42	.51	.52	.48	.50	.45	.39	.42

Table 6: *MUC-4 Results for All Patterns*

that are role-identifying nouns. The Standard patterns in Table 7 were not learned because they did not score highly enough, but the RIE patterns were learned because they performed better. For example, *"<subject> was involved in EVENT"* is a more reliable pattern for identifying perpetrators than just *"<subject> was involved"*. In the disease outbreaks domain, *"<subject> was treated for EVENT"* is more reliable than just *"<subject> was treated"*. Overall, we found many RIE patterns that performed better than their simpler counterparts.

Pattern Type	*Terrorism Perpetrator*	*Pr*
RIE	<subj> was involved in EVENT	.65
standard	<subj> was involved	.32
RIE	<subj> staged EVENT	.27
standard	<subj> staged	.12
RIE	<subj> unleashed EVENT	.33
standard	<subj> unleashed	.17
Pattern Type	*Outbreak Victim*	*Pr*
RIE	<subj> was treated for EVENT	.65
standard	<subj> was treated	.19
RIE	<subj> was hospitalized for EVENT	.75
standard	<subj> was hospitalized	.31
RIE	spread EVENT to <np>	.44
standard	spread to <np>	.10

Table 7: *RIE Patterns vs. Standard Patterns*

5 Related Work

Many supervised learning systems have been developed for event-oriented information extraction (e.g., [13, 2, 5, 6, 4, 3]), but relatively few do not require annotated training data. AutoSlog-TS [10] requires only relevant and irrelevant training documents, and is the baseline system that we used for comparison in our experiments. The systems most similar to ours are ExDisco [17] and Meta-Bootstrapping [11], which are bootstrapping algorithms that require only relevant texts and seed words or patterns for training. However, the extraction patterns produced by Meta-Bootstrapping are general semantic class extractors and not event role extractors. The novel aspects of our work are (1) the use of role-identifying nouns in combination with a semantic bootstrapping algorithm (Basilisk) for extraction pattern learning, and (2) automatically learning a new type of extraction pattern that captures role-identifying expressions.

6 Summary

We have presented a new approach to IE that learns extraction patterns by exploiting role-identifying nouns. We also introduced role-identifying expressions and presented a method for learning them. Our resulting IE system achieved good performance on 7 event roles associated with two different domains.

7 Acknowledgments

This research was supported by Department of Homeland Security Grant N0014-07-1-0152, and the Institute for Scientific Computing Research and the Center for Applied Scientific Computing within Lawrence Livermore National Laboratory. We are grateful to Sean Igo and Rich Warren for annotating the disease outbreaks corpus.

References

[1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM International Conference on Digital Libraries*, 2000.

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

[3] H. Chieu, H. Ng, and Y. Lee. Closing the Gap: Learning-Based Information Extraction Rivaling Knowledge-Engineering Methods. In *ACL-03*, 2003.

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

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

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

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

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

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

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

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

[12] S. Soderland. Learning Information Extraction Rules for Semi-structured and Free Text. *Machine Learning*, 1999.

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

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

[15] M. Thelen and E. Riloff. 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*, pages 214–221, 2002.

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

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