Bootstrapped Training of Event Extraction Classifiers

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

Abstract

Most event extraction systems are trained with supervised learning and rely on a collection of annotated documents. Due to the domain-specificity of this task, event extraction systems must be retrained with new annotated data for each domain. In this paper, we propose a bootstrapping solution for event role filler extraction that requires minimal human supervision. We aim to rapidly train a state-of-the-art event extraction system using a small set of "seed nouns" for each event role, a collection of relevant (in-domain) and irrelevant (out-of-domain) texts, and a semantic dictionary. The experimental results show that the bootstrapped system outperforms previous weakly supervised event extraction systems on the MUC-4 data set, and achieves performance levels comparable to supervised training with 700 manually annotated documents.

1 Introduction

Event extraction systems process stories about domain-relevant events and identify the role fillers of each event. A key challenge for event extraction is that recognizing role fillers is inherently contextual. For example, a PERSON can be a perpetrator or a victim in different contexts (e.g., *"John Smith assassinated the mayor"* vs. *"John Smith was assassinated"*). Similarly, any COMPANY can be an acquirer or an acquiree depending on the context.

Many supervised learning techniques have been used to create event extraction systems using gold standard "answer key" event templates for training (e.g., (Freitag, 1998a; Chieu and Ng, 2002; Maslennikov and Chua, 2007)). However, manually generating answer keys for event extraction is time-consuming and tedious. And more importantly, event extraction annotations are highly domain-specific, so new annotations must be obtained for each domain.

The goal of our research is to use bootstrapping techniques to automatically train a state-of-the-art event extraction system without human-generated answer key templates. The focus of our work is the TIER event extraction model, which is a multi-layered architecture for event extraction (Huang and Riloff, 2011). TIER's innovation over previous techniques is the use of four different classifiers that analyze a document at increasing levels of granularity. TIER progressively zooms in on event information using a pipeline of classifiers that perform document-level classification, sentence classification, and noun phrase classification. TIER outperformed previous event extraction systems on the MUC-4 data set, but relied heavily on a large collection of 1,300 documents coupled with answer key templates to train its four classifiers.

In this paper, we present a bootstrapping solution that exploits a large unannotated corpus for training by using *role-identifying nouns* (Phillips and Riloff, 2007) as seed terms. Phillips and Riloff observed that some nouns, by definition, refer to entities or objects that play a specific role in an event. For example, "assassin", "sniper", and "hitman" refer to people who play the role of PERPETRATOR in a criminal event. Similarly, "victim", "casualty", and "fatality" refer to people who play the role of VICTIM, by virtue of their lexical semantics. Phillips and Riloff called these words *role-identifying nouns* and used them

Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 286–295,
Avignon, France, April 23 - 27 2012.

to learn extraction patterns. Our research also uses *role-identifying nouns* to learn extraction patterns, but the role-identifying nouns and patterns are then used to create training data for event extraction classifiers. Each classifier is then self-trained in a bootstrapping loop.

Our weakly supervised training procedure requires a small set of "seed nouns" for each event role, and a collection of relevant (in-domain) and irrelevant (out-of-domain) texts. No answer key templates or annotated texts are needed. The seed nouns are used to automatically generate a set of *role-identifying patterns*, and then the nouns, patterns, and a semantic dictionary are used to label training instances. We also propagate the event role labels across coreferent noun phrases within a document to produce additional training instances. The automatically labeled texts are used to train three components of TIER: its two types of sentence classifiers and its noun phrase classifiers. To create TIER's fourth component, its document genre classifier, we apply heuristics to the output of the sentence classifiers.

We present experimental results on the MUC-4 data set, which is a standard benchmark for event extraction research. Our results show that the bootstrapped system, TIER$_{lite}$, outperforms previous weakly supervised event extraction systems and achieves performance levels comparable to supervised training with 700 manually annotated documents.

2 Related Work

Event extraction techniques have largely focused on detecting event "triggers" with their arguments for extracting role fillers. Classical methods are either pattern-based (Kim and Moldovan, 1993; Riloff, 1993; Soderland et al., 1995; Huffman, 1996; Freitag, 1998b; Ciravegna, 2001; Califf and Mooney, 2003; Riloff, 1996; Riloff and Jones, 1999; Yangarber et al., 2000; Sudo et al., 2003; Stevenson and Greenwood, 2005) or classifier-based (e.g., (Freitag, 1998a; Chieu and Ng, 2002; Finn and Kushmerick, 2004; Li et al., 2005; Yu et al., 2005)).

Recently, several approaches have been proposed to address the insufficiency of using only local context to identify role fillers. Some approaches look at the broader sentential context around a potential role filler when making a decision (e.g., (Gu and Cercone, 2006; Patwardhan and Riloff, 2009)). Other systems take a more global view and consider discourse properties of the document as a whole to improve performance (e.g., (Maslennikov and Chua, 2007; Ji and Grishman, 2008; Liao and Grishman, 2010; Huang and Riloff, 2011)). Currently, the learning-based event extraction systems that perform best all use supervised learning techniques that require a large number of texts coupled with manually-generated annotations or answer key templates.

A variety of techniques have been explored for weakly supervised training of event extraction systems, primarily in the realm of pattern or rule-based approaches (e.g., (Riloff, 1996; Riloff and Jones, 1999; Yangarber et al., 2000; Sudo et al., 2003; Stevenson and Greenwood, 2005)). In some of these approaches, a human must manually review and "clean" the learned patterns to obtain good performance. Research has also been done to learn extraction patterns in an unsupervised way (e.g., (Shinyama and Sekine, 2006; Sekine, 2006)). But these efforts target open domain information extraction. To extract domain-specific event information, domain experts are needed to select the pattern subsets to use.

There have also been weakly supervised approaches that use more than just local context. (Patwardhan and Riloff, 2007) uses a semantic affinity measure to learn primary and secondary patterns, and the secondary patterns are applied only to event sentences. The event sentence classifier is self-trained using seed patterns. Most recently, (Chambers and Jurafsky, 2011) acquire event words from an external resource, group the event words to form event scenarios, and group extraction patterns for different event roles. However, these weakly supervised systems produce substantially lower performance than the best supervised systems.

3 Overview of TIER

The goal of our research is to develop a weakly supervised training process that can successfully train a state-of-the-art event extraction system for a new domain with minimal human input. We decided to focus our efforts on the TIER event extraction model because it recently produced better performance on the MUC-4 data set than prior learning-based event extraction systems (Huang and Riloff, 2011). In this section, we briefly give an overview of TIER's architecture and its com-

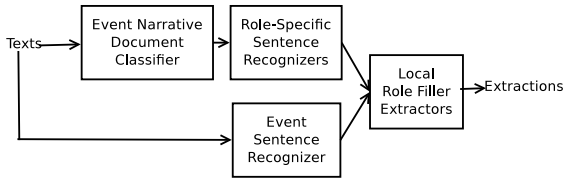

Figure 1: TIER Overview

ponents.

TIER is a multi-layered architecture for event extraction, as shown in Figure 1. Documents pass through a pipeline where they are analyzed at different levels of granularity, which enables the system to gradually "zoom in" on relevant facts. The pipeline consists of a document genre classifier, two types of sentence classifiers, and a set of noun phrase (role filler) classifiers.

The lower pathway in Figure 1 shows that all documents pass through an *event sentence classifier*. Sentences labeled as event descriptions then proceed to the noun phrase classifiers, which are responsible for identifying the role fillers in each sentence. The upper pathway in Figure 1 involves a document genre classifier to determine whether a document is an "event narrative" story (i.e., an article that primarily discusses the details of a domain-relevant event). Documents that are classified as event narratives warrant additional scrutiny because they most likely contain a lot of event information. Event narrative stories are processed by an additional set of *role-specific sentence classifiers* that look for role-specific contexts that will not necessarily mention the event. For example, a victim may be mentioned in a sentence that describes the aftermath of a crime, such as transportation to a hospital or the identification of a body. Sentences that are determined to have "role-specific" contexts are passed along to the noun phrase classifiers for role filler extraction. Consequently, event narrative documents pass through both the lower pathway and the upper pathway. This approach creates an event extraction system that can discover role fillers in a variety of different contexts by considering the type of document being processed.

TIER was originally trained with supervised learning using 1,300 texts and their corresponding answer key templates from the MUC-4 data set (MUC-4 Proceedings, 1992). Human-generated answer key templates are expensive to produce because the annotation process is both difficult and time-consuming. Furthermore, answer key templates for one domain are virtually never reusable for different domains, so a new set of answer keys must be produced from scratch for each domain. In the next section, we present our weakly supervised approach for training TIER's event extraction classifiers.

4 Bootstrapped Training of Event Extraction Classifiers

We adopt a two-phase approach to train TIER's event extraction modules using minimal human-generated resources. The goal of the first phase is to automatically generate positive training examples using *role-identifying* seed nouns as input. The seed nouns are used to automatically generate a set of *role-identifying patterns* for each event role. Each set of patterns is then assigned a set of semantic constraints (selectional restrictions) that are appropriate for that event role. The semantic constraints consist of the role-identifying seed nouns as well as general semantic classes that constrain the event role (e.g., a victim must be a HUMAN). A noun phrase will satisfy the semantic constraints if its head noun is in the seed noun list or if it has the appropriate semantic type (based on dictionary lookup). Each pattern is then matched against the unannotated texts, and if the extracted noun phrase satisfies its semantic constraints, then the noun phrase is automatically labeled as a role filler.

The second phase involves bootstrapped training of TIER's classifiers. Using the labeled instances generated in the first phase, we iteratively train three of TIER's components: the two types of sentential classifiers and the noun phrase classifiers. For the fourth component, the document classifier, we apply heuristics to the output of the sentence classifiers to assess the density of relevant sentences in a document and label high-density stories as event narratives. In the following sections, we present the details of each of these steps.

4.1 Automatically Labeling Training Data

Finding seeding instances of high precision and reasonable coverage is important in bootstrapping. However, this is especially challenging for event extraction task because identifying role fillers is inherently contextual. Furthermore, role

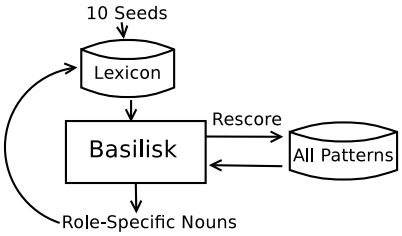

Figure 2: Using Basilisk to Induce Role-Identifying Patterns

fillers occur sparsely in text and in diverse contexts.

In this section, we explain how we generate *role-identifying patterns* automatically using seed nouns, and we discuss why we add semantic constraints to the patterns when producing labeled instances for training. Then, we discuss the coreference-based label propagation that we used to obtain additional training instances. Finally, we give examples to illustrate how we create training instances.

4.1.1 Inducing Role-Identifying Patterns

The input to our system is a small set of manually-defined *seed nouns* for each event role. Specifically, the user is required to provide 10 *role-identifying nouns* for each event role. (Phillips and Riloff, 2007) defined a noun as being "role-identifying" if its lexical semantics reveal the role of the entity/object in an event. For example, the words "assassin" and "sniper" are people who participate in a violent event as a PERPETRATOR. Therefore, the entities referred to by role-identifying nouns are probable role fillers.

However, treating every context surrounding a role-identifying noun as a role-identifying pattern is risky. The reason is that many instances of role-identifying nouns appear in contexts that do not describe the event. But, if one pattern has been seen to extract many role-identifying nouns and seldomly seen to extract other nouns, then the pattern likely represents an event context.

As (Phillips and Riloff, 2007) did, we use Basilisk to learn patterns for each event role. Basilisk was originally designed for semantic class learning (e.g., to learn nouns belonging to semantic categories, such as *building* or *human*). As shown in Figure 2, beginning with a small set of seed nouns for each semantic class, Basilisk learns additional nouns belonging to the same semantic class. Internally, Basilisk uses extraction patterns automatically generated from unannotated texts to assess the similarity of nouns. First, Basilisk assigns a score to each pattern based on the number of seed words that co-occur with it. Basilisk then collects the noun phrases extracted by the highest-scoring patterns. Next, the head noun of each noun phrase is assigned a score based on the set of patterns that it co-occurred with. Finally, Basilisk selects the highest-scoring nouns, automatically labels them with the semantic class of the seeds, adds these nouns to the lexicon, and restarts the learning process in a bootstrapping fashion.

For our work, we give Basilisk *role-identifying seed nouns* for each event role. We run the bootstrapping process for 20 iterations and then harvest the 40 best patterns that Basilisk identifies for each event role. We also tried using the additional role-identifying nouns learned by Basilisk, but found that these nouns were too noisy.

4.1.2 Using the Patterns to Label NPs

The induced *role-identifying patterns* can be matched against the unannotated texts to produce labeled instances. However, relying solely on the pattern contexts can be misleading. For example, the pattern context <*subject*> *caused damage* will extract some noun phrases that are weapons (e.g., *the bomb*) but some noun phrases that are not (e.g., *the tsunami*).

Based on this observation, we add selectional restrictions to each pattern that requires a noun phrase to satisfy certain semantic constraints in order to be extracted and labeled as a positive instances for an event role. The selectional restrictions are satisfied if the head noun is among the *role-identifying seed nouns* or if the semantic class of the head noun is compatible with the corresponding event role. In the previous example, *tsunami* will not be extracted as a weapon because it has an incompatible semantic class (EVENT), but *bomb* will be extracted because it has a compatible semantic class (WEAPON).

We use the semantic class labels assigned by the Sundance parser (Riloff and Phillips, 2004) in our experiments. Sundance looks up each noun in a semantic dictionary to assign the semantic class labels. As an alternative, general resources (e.g., WordNet (Miller, 1990)) or a semantic tagger (e.g., (Huang and Riloff, 2010)) could be used.

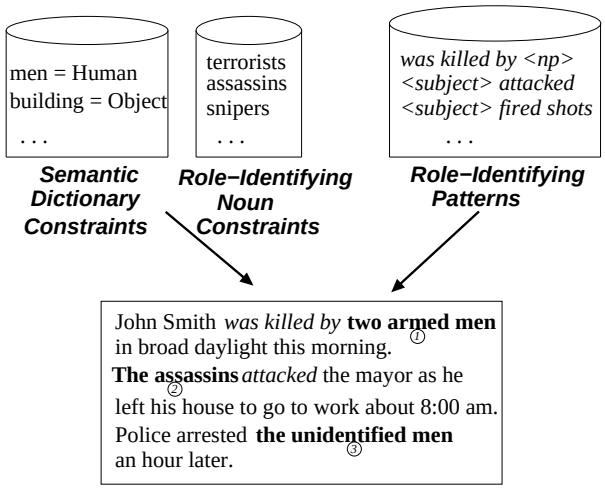

Figure 3: Automatic Training Data Creation

4.1.3 Propagating Labels with Coreference

To enrich the automatically labeled training instances, we also propagate the event role labels across coreferent noun phrases within a document. The observation is that once a noun phrase has been identified as a role filler, its coreferent mentions in the same document likely fill the same event role since they are referring to the same real world entity.

To leverage these coreferential contexts, we employ a simple head noun matching heuristic to identify coreferent noun phrases. This heuristic assumes that two noun phrases that have the same head noun are coreferential. We considered using an off-the-shelf coreference resolver, but decided that the head noun matching heuristic would likely produce higher precision results, which is important to produce high-quality labeled data.

4.1.4 Examples of Training Instance Creation

Figure 3 illustrates how we label training instances automatically. The text example shows three noun phrases that are automatically labeled as perpetrators. Noun phrases #1 and #2 occur in role-identifying pattern contexts (*was killed by* <*np*> and <*subject*> *attacked*) and satisfy the semantic constraints for perpetrators because "men" has a compatible semantic type and "assassins" is a role-identifying noun for perpetrators.

Noun phrase #3 ("the unidentified men") does not occur in a pattern context, but it is deemed to be coreferent with "two armed men" because they have the same head noun. Consequently, we propagate the perpetrator label from noun phrase #1 to noun phrase #3.

4.2 Creating TIER$_{lite}$ with Bootstrapping

In this section, we explain how the labeled instances are used to train TIER's classifiers with bootstrapping. In addition to the automatically labeled instances, the training process depends on a text corpus that consists of both relevant (in-domain) and irrelevant (out-of-domain) documents. Positive instances are generated from the relevant documents and negative instances are generated by randomly sampling from the irrelevant documents.

The classifiers are all support vector machines (SVMs), implemented using the SVMlin software (Keerthi and DeCoste, 2005). When applying the classifiers during bootstrapping, we use a sliding confidence threshold to determine which labels are reliable based on the values produced by the SVM. Initially, we set the threshold to be 2.0 to identify highly confident predictions. But if fewer than k instances pass the threshold, then we slide the threshold down in decrements of 0.1 until we obtain at least k labeled instances or the threshold drops below 0, in which case bootstrapping ends. We used k=10 for both sentence classifiers and k=30 for the noun phrase classifiers.

The following sections present the details of the bootstrapped training process for each of TIER's components.

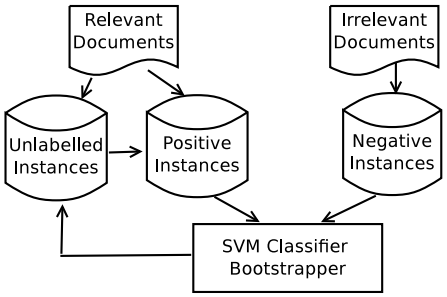

Figure 4: The Bootstrapping Process

4.2.1 Noun Phrase Classifiers

The mission of the noun phrase classifiers is to determine whether a noun phrase is a plausible event role filler based on the local features surrounding the noun phrase (NP). A set of classifiers is needed, one for each event role.

As shown in Figure 4, to seed the classifier training, the positive noun phrase instances are

generated from the relevant documents following Section 4.1. The negative noun phrase instances are drawn randomly from the irrelevant documents. Considering the sparsity of role fillers in texts, we set the negative:positive ratio to be 10:1. Once the classifier is trained, it is applied to the unlabeled noun phrases in the relevant documents. Noun phrases that are assigned role filler labels by the classifier with high confidence (using the sliding threshold) are added to the set of positive instances. New negative instances are drawn randomly from the irrelevant documents to maintain the 10:1 (negative:positive) ratio.

We extract features from each noun phrase (NP) and its surrounding context. The features include the NP head noun and its premodifiers. We also use the Stanford NER tagger (Finkel et al., 2005) to identify Named Entities within the NP. The context features include four words to the left of the NP, four words to the right of the NP, and the lexico-syntactic patterns generated by AutoSlog to capture expressions around the NP (see (Riloff, 1993) for details).

4.2.2 Event Sentence Classifier

The event sentence classifier is responsible for identifying sentences that describe a relevant event. Similar to the noun phrase classifier training, positive training instances are selected from the relevant documents and negative instances are drawn from the irrelevant documents. All sentences in the relevant documents that contain one or more labeled noun phrases (belonging to any event role) are labeled as positive training instances. We randomly sample sentences from the irrelevant documents to obtain a negative:positive training instance ratio of 10:1. The bootstrapping process is then identical to that of the noun phrase classifiers. The feature set for this classifier consists of unigrams, bigrams and AutoSlog's lexico-syntactic patterns surrounding all noun phrases in the sentence.

4.2.3 Role-Specific Sentence Classifiers

The role-specific sentence classifiers are trained to identify the contexts specific to each event role. All sentences in the relevant documents that contain at least one labeled noun phrase for the appropriate event role are used as positive instances. Negative instances are randomly sampled from the irrelevant documents to maintain the negative:positive ratio of 10:1. The bootstrapping process and feature set are the same as for the event sentence classifier.

The difference between the two types of sentence classifiers is that the event sentence classifier uses positive instances from all event roles, while each role-specific sentence classifiers only uses the positive instances for one particular event role. The rationale is similar as in the supervised setting (Huang and Riloff, 2011); the event sentence classifier is expected to generalize over all event roles to identify event mention contexts, while the role-specific sentence classifiers are expected to learn to identify contexts specific to individual roles.

4.2.4 Event Narrative Document Classifier

TIER also uses an event narrative document classifier and only extracts information from role-specific sentences within event narrative documents. In the supervised setting, TIER uses heuristic rules derived from answer key templates to identify the event narrative documents in the training set, which are used to train an event narrative document classifier. The heuristic rules require that an event narrative should have a high density of relevant information and tend to mention the relevant information within the first several sentences.

In our weakly supervised setting, we use the information density heuristic directly instead of training an event narrative classifier. We approximate the relevant information density heuristic by computing the ratio of relevant sentences (both event sentences and role-specific sentences) out of all the sentences in a document. Thus, the event narrative labeller only relies on the output of the two sentence classifiers. Specifically, we label a document as an event narrative if $\geq 50\%$ of the sentences in the document are relevant (i.e., labeled positively by either sentence classifier).

5 Evaluation

In this section, we evaluate our bootstrapped system, TIER$_{lite}$, on the MUC-4 event extraction data set. First, we describe the IE task, the data set, and the weakly supervised baseline systems that we use for comparison. Then we present the results of our fully bootstrapped system TIER$_{lite}$, the weakly supervised baseline systems, and two fully supervised event extraction systems, TIER

and GLACIER. In addition, we analyze the performance of TIER$_{lite}$ using different configurations to assess the impact of its components.

5.1 IE Task and Data

We evaluated the performance of our systems on the MUC-4 terrorism IE task (MUC-4 Proceedings, 1992) about Latin American terrorist events. We used 1,300 texts (DEV) as our training set and 200 texts (TST3+TST4) as the test set. All the documents have answer key templates. For the training set, we used the answer keys to separate the documents into relevant and irrelevant subsets. Any document containing at least one relevant event was considered to be relevant.

PerpInd	PerpOrg	Target	Victim	Weapon
129	74	126	201	58

Table 1: # of Role Fillers in the MUC-4 Test Set

Following previous studies, we evaluate our system on five MUC-4 string event roles: *perpetrator individuals* (PerpInd), *perpetrator organizations* (PerpOrg), *physical targets*, *victims*, and *weapons*. Table 1 shows the distribution of role fillers in the MUC-4 test set. The complete IE task involves the creation of answer key templates, one template per event[1]. Our work focuses on extracting individual role fillers and not template generation, so we evaluate the accuracy of the role fillers irrespective of which template they occur in.

We used the same *head noun* scoring scheme as previous systems, where an extraction is correct if its head noun matches the head noun in the answer key[2]. Pronouns were discarded from both the system responses and the answer keys since no coreference resolution is done. Duplicate extractions were conflated before being scored, so they count as just one hit or one miss.

5.2 Weakly Supervised Baselines

We compared the performance of our system with three previous weakly supervised event extraction systems.

AutoSlog-TS (Riloff, 1996) generates lexico-syntactic patterns exhaustively from unannotated texts and ranks them based on their frequency and probability of occurring in relevant documents. A human expert then examines the patterns and manually selects the best patterns for each event role. During testing, the patterns are matched against unseen texts to extract event role fillers.

PIPER (Patwardhan and Riloff, 2007; Patwardhan, 2010) learns extraction patterns using a semantic affinity measure, and it distinguishes between primary and secondary patterns and applies them selectively. (Chambers and Jurafsky, 2011) (C+J) created an event extraction system by acquiring event words from WordNet (Miller, 1990), clustering the event words into different event scenarios, and grouping extraction patterns for different event roles.

5.3 Performance of TIER$_{lite}$

Table 2 shows the seed nouns that we used in our experiments, which were generated by sorting the nouns in the corpus by frequency and manually identifying the first 10 role-identifying nouns for each event role.[3] Table 3 shows the number of training instances (noun phrases) that were automatically labeled for each event role using our training data creation approach (Section 4.1).

Event Role	**Seed Nouns**
Perpetrator Individual	terrorists assassins criminals rebels murderers death_squads guerrillas member members individuals
Perpetrator Organization	FMLN ELN FARC MRTA M-19 Front Shining_Path Medellin_Cartel The_Extraditables Army_of_National_Liberation
Target	houses residence building home homes offices pipeline hotel car vehicles
Victim	victims civilians children jesuits Galan priests students women peasants Romero
Weapon	weapons bomb bombs explosives rifles dynamite grenades device car_bomb

Table 2: Role-Identifying Seed Nouns

PerpInd	PerpOrg	Target	Victim	Weapon
296	157	522	798	248

Table 3: # of Automatically Labeled NPs

Table 4 shows how our bootstrapped system TIER$_{lite}$ compares with previous weakly supervised systems and two supervised systems, its supervised counterpart TIER (Huang and Riloff, 2011) and a model that jointly considers local and sentential contexts, GLACIER (Patwardhan

[1]Documents may contain multiple events per article.

[2]For example, "armed men" will match "5 armed men".

[3]We only found 9 weapon terms among the high-frequency terms.

Weakly Supervised Baselines						
	PerpInd	PerpOrg	Target	Victim	Weapon	Average
AUTOSLOG-TS (1996)	33/49/40	52/33/41	54/59/56	49/54/51	38/44/41	45/48/46
PIPER$_{Best}$ (2007)	39/48/43	55/31/40	37/60/46	44/46/45	47/47/47	44/46/45
C+J (2011)	-	-	-	-	-	44/36/40
Supervised Models						
GLACIER (2009)	51/58/54	34/45/38	43/72/53	55/58/56	57/53/55	48/57/52
TIER (2011)	48/57/52	46/53/50	51/73/60	56/60/58	53/64/58	51/62/56
Weakly Supervised Models						
TIER$_{lite}$	47/51/49	60/39/47	37/65/47	39/53/45	53/55/54	47/53/50

Table 4: Performance of the Bootstrapped Event Extraction System (Precision/Recall/F-score)

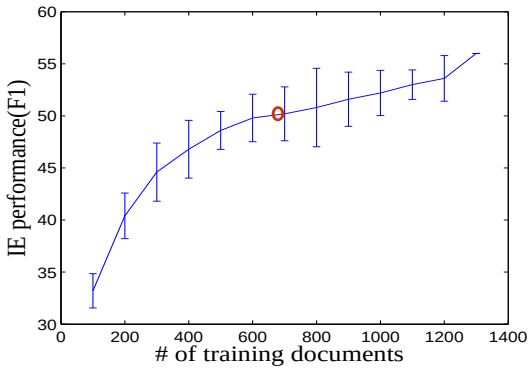

Figure 5: The Learning Curve of Supervised TIER

and Riloff, 2009). We see that TIER$_{lite}$ outperforms all three weakly supervised systems, with slightly higher precision and substantially more recall. When compared to the supervised systems, the performance of TIER$_{lite}$ is similar to GLACIER, with comparable precision but slightly lower recall. But the supervised TIER system, which was trained with 1,300 annotated documents, is still superior, especially in recall.

Figure 5 shows the learning curve for TIER when it is trained with fewer documents, ranging from 100 to 1,300 in increments of 100. Each data point represents five experiments where we randomly selected k documents from the training set and averaged the results. The bars show the range of results across the five runs. Figure 5 shows that TIER's performance increases from an F score of 34 when trained on just 100 documents up to an F score of 56 when training on 1,300 documents. The circle shows the performance of our bootstrapped system, TIER$_{lite}$, which achieves an F score comparable to supervised training with about 700 manually annotated documents.

5.4 Analysis

Table 6 shows the effect of the coreference propagation step described in Section 4.1.3 as part of training data creation. Without this step, the performance of the bootstrapped system yields an F score of 41. With the benefit of the additional training instances produced by coreference propagation, the system yields an F score of 53. The new instances produced by coreference propagation seem to substantially enrich the diversity of the set of labeled instances.

Seeding	P/R/F
wo/Coref	45/38/41
w/Coref	47/53/50

Table 6: Effects of Coreference Propagation

In the evaluation section, we saw that the supervised event extraction systems achieve higher recall than the weakly supervised systems. Although our bootstrapped event extraction system TIER$_{lite}$ produces higher recall than previous weakly supervised systems, a substantial recall gap still exists.

Considering the pipeline structure of the event extraction system, as shown in Figure 1, the noun phrase extractors are responsible for identifying all candidate role fillers. The sentential classifiers and the document classifier effectively serve as filters to rule out candidates from irrelevant contexts. Consequently, there is no way to recover missing recall (role fillers) if the noun phrase extractors fail to identify them.

Since the noun phrase classifiers are so central to the performance of the system, we compared the performance of the bootstrapped noun phrase classifiers directly with their supervised conterparts. The results are shown in Table 5. Both sets of classifiers produce low precision when used in isolation, but their precision levels are compara-

	PerpInd	PerpOrg	Target	Victim	Weapon	Average
Supervised Classifier	25/67/36	26/78/39	34/83/49	32/72/45	30/75/43	30/75/42
Bootstrapped Classifier	30/54/39	37/53/44	30/71/42	28/63/39	36/57/44	32/60/42

Table 5: Evaluation of Bootstrapped Noun Phrase Classifiers (Precision/Recall/F-score)

ble. The TIER pipeline architecture is successful at eliminating many of the false hits. However, the recall of the bootstrapped classifiers is consistently lower than the recall of the supervised classifiers. Specifically, the recall is about 10 points lower for three event roles (*PerpInd*, *Target* and *Victim*) and 20 points lower for the other two event roles (*PerpOrg* and *Weapon*). These results suggest that our bootstrapping approach to training instance creation does not fully capture the diversity of role filler contexts that are available in the supervised training set of 1,300 documents. This issue is an interesting direction for future work.

6 Conclusions

We have presented a bootstrapping approach for training a multi-layered event extraction model using a small set of "seed nouns" for each event role, a collection of relevant (in-domain) and irrelevant (out-of-domain) texts and a semantic dictionary. The experimental results show that the bootstrapped system, TIER$_{lite}$, outperforms previous weakly supervised event extraction systems on a standard event extraction data set, and achieves performance levels comparable to supervised training with 700 manually annotated documents. The minimal supervision required to train such a model increases the portability of event extraction systems.

7 Acknowledgments

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

References

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

Nathanael Chambers and Dan Jurafsky. 2011. Template-Based Information Extraction without the Templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-11)*.

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

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

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

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

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

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

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

Ruihong Huang and Ellen Riloff. 2010. Inducing Domain-specific Semantic Class Taggers from (Almost) Nothing. In *Proceedings of The 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*.

Ruihong Huang and Ellen Riloff. 2011. Peeling Back the Layers: Detecting Event Role Fillers in Secondary Contexts. In *Proceedings of the 49th Annual*

Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-11).

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

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

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

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

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

Shasha Liao and Ralph Grishman. 2010. Using Document Level Cross-Event Inference to Improve Event Extraction. In *Proceedings of the 48st Annual Meeting on Association for Computational Linguistics (ACL-10)*.

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

G. Miller. 1990. Wordnet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4).

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

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

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

S. Patwardhan. 2010. *Widening the Field of View of Information Extraction through Sentential Event Recognition*. Ph.D. thesis, University of Utah.

W. Phillips 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)*, pages 468–473.

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

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

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

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

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

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

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

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

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

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

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