Multi-faceted Event Recognition with Bootstrapped Dictionaries

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

Abstract

Identifying documents that describe a specific type of event is challenging due to the high complexity and variety of event descriptions. We propose a *multi-faceted event recognition* approach, which identifies documents about an event using event phrases as well as defining characteristics of the event. Our research focuses on civil unrest events and learns civil unrest expressions as well as phrases corresponding to potential agents and reasons for civil unrest. We present a bootstrapping algorithm that automatically acquires event phrases, agent terms, and purpose (reason) phrases from unannotated texts. We use the bootstrapped dictionaries to identify civil unrest documents and show that multi-faceted event recognition can yield high accuracy.

1 Introduction

Many people are interested in following news reports about events. Government agencies are keenly interested in news about civil unrest, acts of terrorism, and disease outbreaks. Companies want to stay on top of news about corporate acquisitions, high-level management changes, and new joint ventures. The general public is interested in articles about crime, natural disasters, and plane crashes. We will refer to the task of identifying documents that describe a specific type of event as *event recognition*.

It is tempting to assume that event keywords are sufficient to identify documents that discuss instances of an event. But event words are rarely reliable on their own. For example, consider the challenge of finding documents about civil unrest. The words *"strike"*, *"rally"*, and *"riot"* refer to common types of civil unrest, but they frequently refer to other things as well. A strike can refer to a military event or a sporting event (e.g., *"air strike"*, *"bowling strike"*), a rally can be a race or a spirited exchange (e.g.,*"car rally"*, *"tennis rally"*), and a riot can refer to something funny (e.g., *"she's a riot"*). Event keywords also appear in general discussions that do not mention a specific event (e.g., *"37 states prohibit teacher strikes"* or *"The fine for inciting a riot is $1,000"*). Furthermore, many relevant documents are not easy to recognize because events can be described with complex expressions that do not include event keywords. For example, *"took to the streets"*, *"walked off their jobs"* and *"stormed parliament"* often describe civil unrest.

The goal of our research is to recognize event descriptions in text by identifying event expressions as well as defining characteristics of the event. We propose that *agents* and *purpose* are characteristics of an event that are essential to distinguish one type of event from another. The agent responsible for an action often determines how we categorize the action. For example, natural disasters, military operations, and terrorist attacks can all produce human casualties and physical destruction. But the agent of a natural disaster must be a natural force, the agent of a military incident must be military personnel, and the agent of a terrorist attack is never a natural force and rarely military personnel. There may be other important factors as well, but the agent is often an essential part of an event definition.

The purpose of an event is also a crucial factor in distinguishing between event types. For exam-

Proceedings of NAACL-HLT 2013, pages 41–51,
Atlanta, Georgia, 9–14 June 2013.

ple, civil unrest events and sporting events both involve large groups of people amassing at a specific site. But the purpose of civil unrest gatherings is to protest against socio-political problems, while sporting events are intended as entertainment. As another example, terrorist events and military incidents can both cause casualties, but the purpose of terrorism is to cause widespread fear, while the purpose of military actions is to protect national security interests.

Our research explores the idea of *multi-faceted event recognition*: using event expressions as well as facets of the event (agents and purpose) to identify documents about a specific type of event. We present a bootstrapping framework to automatically create event phrase, agent, and purpose dictionaries. The learning process uses unannotated texts, a few event keywords, and seed terms for common agents and purpose phrases associated with the event type.

Our bootstrapping algorithm exploits the observation that event expressions, agents, and purpose phrases often appear together in sentences that introduce an event. In the first step, we extract event expressions based on dependency relations with an agent and purpose phrase. The harvested event expressions are added to an event phrase dictionary. In the second step, new agent terms are extracted from sentences containing an event phrase and a purpose phrase, and new purpose phrases are harvested from sentences containing an event phrase and an agent. These harvested terms are added to agent and purpose dictionaries. The bootstrapping algorithm ricochets back and forth, alternately learning new event phrases and learning new agent/purpose phrases, in an iterative process.

We explore several ways of using these bootstrapped dictionaries. We conclude that finding at least two different types of event information produces high accuracy (88% precision) with good recall (71%) on documents that contain an event keyword. We also present experiments with documents that do not contain event keywords, and obtain 74% accuracy when matching all three types of event information.

2 Related Work

Event recognition has been studied in several different contexts. There has been a lot of research on event extraction, where the goal is to extract facts about events from text (e.g., (ACE Evaluations, 2006; Appelt et al., 1993; Riloff, 1996; Yangarber et al., 2000; Chieu and Ng, 2002; Califf and Mooney, 2003; Sudo et al., 2003; Stevenson and Greenwood, 2005; Sekine, 2006)). Although our research does not involve extracting facts, event extraction systems can also be used to identify stories about a specific type of event. For example, the MUC-4 evaluation (MUC-4 Proceedings, 1992) included "text filtering" results that measured the performance of event extraction systems at identifying event-relevant documents. The best text filtering results were high (about 90% F score), but relied on hand-built event extraction systems. More recently, some research has incorporated event region detectors into event extraction systems to improve extraction performance (Gu and Cercone, 2006; Patwardhan and Riloff, 2007; Huang and Riloff, 2011).

There has been recent work on event detection from social media sources (Becker et al., 2011; Popescu et al., 2011). Some research identifies specific types of events in tweets, such as earthquakes (Sakaki et al., 2010) and entertainment events (Benson et al., 2011). There has also been work on event trend detection (Lampos et al., 2010; Mathioudakis and Koudas, 2010) and event prediction through social media, such as predicting elections (Tumasjan et al., 2010; Conover et al., 2011) or stock market indicators (Zhang et al., 2010). (Ritter et al., 2012) generated a calendar of events mentioned on twitter. (Metzler et al., 2012) proposed structured retrieval of historical event information over microblog archives by distilling high quality event representations using a novel temporal query expansion technique.

Some text classification research has focused on event categories. (Riloff and Lehnert, 1994) used an information extraction system to generate *relevancy signatures* that were indicative of different event types. This work originally relied on manually labeled patterns and a hand-crafted semantic dictionary. Later work (Riloff and Lorenzen, 1999) eliminated the need for the dictionary and labeled patterns, but still assumed the availability of relevant/irrelevant training texts.

Event recognition is also related to Topic Detection and Tracking (TDT) (Allan et al., 1998; Allan,

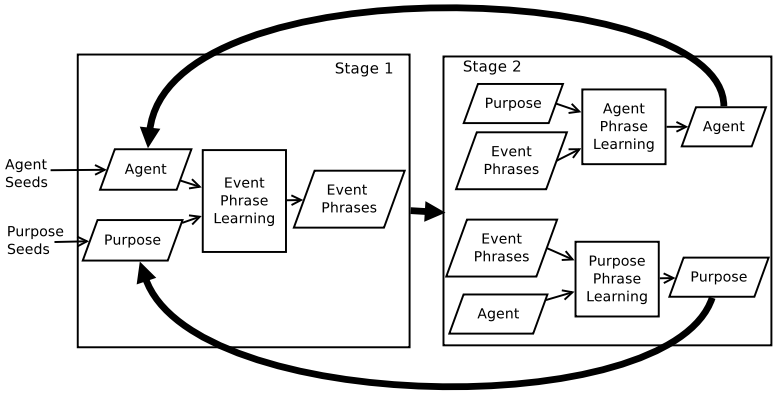

Figure 1: Bootstrapped Learning of Event Dictionaries

2002) which addresses event-based organization of a stream of news stories. Event recognition is similar to New Event Detection, also called First Story Detection, which is considered the most difficult TDT task (Allan et al., 2000a). Typical approaches reduce documents to a set of features, either as a word vector (Allan et al., 2000b) or a probability distribution (Jin et al., 1999), and compare the incoming stories to stories that appeared in the past by computing similarities between their feature representations. Recently, event paraphrases (Petrovic et al., 2012) have been explored to deal with the diversity of event descriptions. However, the New Event Detection task differs from our event recognition task because we want to find all stories describing a certain type of event, not just new events.

3 Bootstrapped Learning of Event Dictionaries

Our bootstrapping approach consists of two stages of learning as shown in Figure 1. The process begins with a few agent seeds, purpose phrase patterns, and unannotated articles selected from a broad-coverage corpus using event keywords. In the first stage, event expressions are harvested from the sentences that have both an agent and a purpose phrase in specific syntactic positions. In the second stage, new purpose phrases are harvested from sentences that contain both an event phrase and an agent, while new agent terms are harvested from sentences that contain both an event phrase and a purpose phrase. The new terms are added to growing event dictionaries, and the bootstrapping process repeats. Our work focuses on civil unrest events.

3.1 Stage 1: Event Phrase Learning

We first extract potential civil unrest stories from the English Gigaword corpus (Parker et al., 2011) using six civil unrest keywords. As explained in Section 1, event keywords are not sufficient to obtain relevant documents with high precision, so the extracted stories are a mix of relevant and irrelevant articles. Our algorithm first selects sentences to use for learning, and then harvests event expressions from them.

3.1.1 Event Sentence Identification

The input in stage 1 consists of a few agent terms and purpose patterns for seeding. The agent seeds are single nouns, while the purpose patterns are verbs in infinitive or present participle forms. Table 1 shows the agent terms and purpose phrases used in our experiments. The agent terms were manually selected by inspecting the most frequent nouns in the documents with civil unrest keywords. The purpose patterns are the most common verbs that describe the reason for a civil unrest event. We identify *probable event sentences* by extracting all sentences that contain at least one agent term and one purpose phrase.

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

Table 1: Agent and Purpose Phrases Used for Seeding

3.1.2 Harvesting Event Expressions

To constrain the learning process, we require event expressions and purpose phrases to match certain syntactic structures. We apply the Stanford dependency parser (Marneffe et al., 2006) to the probable event sentences to identify verb phrase candidates and to enforce syntactic constraints between the different types of event information.

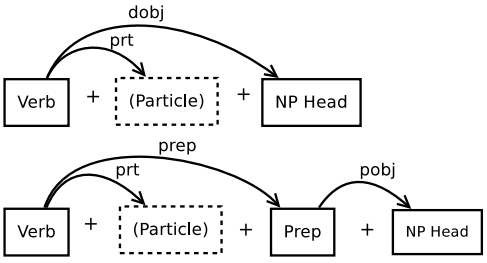

Figure 2: Phrasal Structure of Event & Purpose Phrases

Figure 2 shows the two types of verb phrases that we learn. One type consists of a verb paired with the head noun of its direct object. For example, event phrases can be *"stopped work"* or *"occupied offices"*, and purpose phrases can be *"show support"* or *"condemn war"*. The second type consists of a verb and an attached prepositional phrase, retaining only the head noun of the embedded noun phrase. For example, *"took to street"* and *"scuffled with police"* can be event phrases, while *"call for resignation"* and *"press for wages"* can be purpose phrases. In both types of verb phrases, a particle can optionally follow the verb.

Event expressions, agents, and purpose phrases must appear in specific dependency relations, as illustrated in Figure 3. An agent must be the syntactic subject of the event phrase. A purpose phrase must be a complement of the event phrase, specifically, we require a particular dependency relation, "xcomp"[1], between the two verb phrases. For example, in the sentence *"Leftist activists took to the streets in the Nepali capital Wednesday protesting higher fuel prices."*, the dependency relation "xcomp" links *"took to the streets"* with *"protesting higher fuel prices"*.

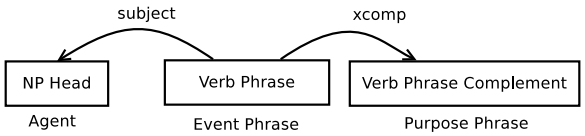

Figure 3: Syntactic Dependencies between Agents, Event Phrases, and Purpose Phrases

Given the syntactic construction shown in Figure 3, with a known agent and purpose phrase, we extract the head verb phrase of the "xcomp" dependency relation as an event phrase candidate. The event phrases that co-occur with at least two unique agent terms and two unique purposes phrases are saved in our event phrase dictionary.

3.2 Stage 2: Learning Agent and Purpose Phrases

In the second stage of bootstrapping, we learn new agent terms and purpose phrases. Our rationale is that if a sentence contains an event phrase and one other important facet of the event (agent or purpose), then the sentence probably describes a relevant event. We can then look for additional facets of the event in the same sentence. We learn both agent and purpose phrases simultaneously in parallel learning processes. As before, we first identify probable event sentences and then harvest agent and purpose phrases from these sentences.

3.2.1 Event Sentence Identification

We identify probable event sentences by extracting sentences that contain at least one event phrase (based on the dictionary produced in the first stage of bootstrapping) and an agent term or a purpose phrase. As before, the event information must occur in the sentential dependency structures shown in Figure 3.

3.2.2 Harvesting Agent and Purpose Phrases

The sentences that contain an event phrase and an agent are used to harvest more purpose phrases, while the sentences that contain an event phrase and a purpose phrase are used to harvest more agent terms. Purpose phrases are extracted from the phrasal structures shown in Figure 2. In the learning process for agents, if a sentence has an event

[1]In the dependency parser, "xcomp" denotes a general relation between a VP or an ADJP and its open clausal complement. For example, in the sentence *"He says that you like to swim."*, the "xcomp" relation will link "like" (head) and "swim" (dependent). With our constraints on the verb phrase forms, the dependent verb phrase in this construction tends to describe the purpose of the verb phrase.

phrase as the head of the "xcomp" dependency relation and a purpose phrase as the dependent clause of the "xcomp" dependency relation, then the head noun of the syntactic subject of the event phrase is harvested as a candidate agent term. We also record the modifiers appearing in all of the noun phrases headed by an agent term. Agent candidates that co-occur with at least two unique event phrases and at least two different modifiers of known agent terms are selected as new agent terms.

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

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

3.3 Domain Relevance Criteria

To avoid domain drift during bootstrapping, we use two additional criteria to discard phrases that are not necessarily associated with the domain.

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

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

We randomly sampled 10% of the Gigaword texts that contain a civil unrest event keyword to create the "domain-specific" corpus, and randomly sampled 10% of the remaining Gigaword texts to create the "broad-coverage" corpus.[2] Keyword-based sampling is an approximation to domain-relevance, but gives us a general idea about the prevalance of a phrase in different types of texts.

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

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

In our experiments we required event and purpose phrases to have *domain-specificity* $\geq .33$ and agent terms to have *event-specificity* $\geq .01$.[3]

4 Evaluation

4.1 Data

We conducted experiments to evaluate the performance of our bootstrapped event dictionaries for recognizing civil unrest events. Civil unrest is a broad term typically used by the media or law enforcement to describe a form of public disturbance that involves a group of people, usually to protest or promote a cause. Civil unrest events include strikes, protests, occupations, rallies, and similar forms of obstructions or riots. We chose six *event keywords* to identify potential civil unrest documents: "protest", "strike", "march", "rally", "riot" and "occupy". We extracted documents from the English Gigaword corpus (Parker et al., 2011) that contain at least one of these event keywords, or a morphological variant of a keyword.[4] This process extracted nearly one million documents, which we will refer to as our *event-keyword corpus*.

We randomly sampled 400 documents[5] from the event-keyword corpus and asked two annotators to determine whether each document mentioned a civil

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

[3]This value is so small because we simply want to filter phrases that virtually never occur in the event sentences, and we can recognize very few event sentences in the early stages of bootstrapping.

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

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

unrest event. We defined annotation guidelines and conducted an inter-annotator agreement study on 100 of these documents. The annotators achieved a κ score of .82. We used these 100 documents as our *tuning set*. Then each annotator annotated 150 more documents to create our *test set* of 300 documents.

4.2 Baselines

The first row of Table 2 shows event recognition accuracy when only the event keywords are used. All of our documents were obtained by searching for a keyword, but only 101 of the 300 documents in our test set were labeled as relevant by the annotators (i.e., 101 describe a civil unrest event). This means that using only the event keywords to identify civil unrest documents yields about 34% precision. In a second experiment, **KeywordTitle**, we required the event keyword to be in the title (headline) of the document. The KeywordTitle approach produced better precision (66%), but only 33% of the relevant documents had a keyword in the title.

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

Table 2: Experimental Results

The second section of Table 2 shows the results of two supervised classifiers based on 10-fold cross validation with our test set. Both classifiers were trained using support vector machines (SVMs) (Joachims, 1999) with a linear kernel (Keerthi and DeCoste, 2005). The first classifier used unigrams as features, while the second classifier used both unigrams and bigrams. All the features are binary. The evaluation results show that the unigram classifier has an F-score of .64. Using both unigram and bigram features increased precision to 71% but recall fell by 7%, yielding a slightly lower F-score of .62.

4.3 Event Recognition with Bootstrapped Dictionaries

Next, we used our bootstrapped dictionaries for event recognition. The bootstrapping process ran for 8 iterations and then stopped because no more phrases could be learned. The quality of bootstrapped data often degrades as bootstrapping progresses, so we used the tuning set to evaluate the dictionaries after each iteration. The best performance[6] on the tuning set resulted from the dictionaries produced after four iterations, so we used these dictionaries for our experiments. Table 3 shows the

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

Table 3: Dictionary Sizes after Several Iterations

number of event phrases, agents and purpose phrases learned after each iteration. All three lexicons were significantly enriched after each iteration. The final bootstrapped dictionaries contain *623* event phrases, *569* purpose phrases and *139* agent terms. Table 4 shows samples from each event dictionary.

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

Table 4: Examples of Dictionary Entries

The third section of Table 2 shows the results when using the bootstrapped dictionaries for event recognition. We used a simple dictionary look-up approach that searched for dictionary entries in each document. Our phrases were generated based on

[6] Based on the performance for the **All Pairs** approach.

syntactic analysis and only head words were retained for generality. But we wanted to match dictionary entries without requiring syntactic analysis of new documents. So we used an approximate matching scheme that required each word to appear within 5 words of the previous word. For example, "held protest" would match "held a large protest" and "held a very large political protest". In this way, we avoid the need for syntactic analysis when using the dictionaries for event recognition.

First, we labeled a document as relevant if it contained any Event Phrase (EV) in our dictionary. The event phrases achieved better performance than all of the baselines, yielding an F-score of 69%. The best baseline was the unigram classifier, which was trained with supervised learning. The bootstrapped event phrase dictionary produced much higher precision (79% vs. 66%) with only slightly lower recall (60% vs. 62%), and did not require annotated texts for training. Statistical significance testing shows that the Event Phrase lookup approach works significantly better than the unigram classifier ($p < 0.05$, paired bootstrap (Berg-Kirkpatrick et al., 2012)).

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

Our original hypothesis was that a single type of event information is not sufficient to accurately identify event descriptions. Our goal was high-accuracy event recognition by requiring that a document contain multiple clues pertaining to different facets of an event (*multi-faceted event recognition*). The last row of Table 2 (**All Pairs**) shows the results when requiring matches from at least two different bootstrapped dictionaries. Specifically, we labeled a document as relevant if it contained at least one phrase from each of two different dictionaries and these phrases occurred in the same sentence. Table 2 shows that multi-faceted event recognition achieves 88% precision with reasonably good recall of 71%, yielding an F-score of 79%. This multi-faceted approach with simple dictionary look-up outperformed all of the baselines, and each dictionary used by itself. Statistical significance testing shows that the All Pairs approach works significantly better than the unigram classifier ($p < 0.001$, paired bootstrap). The All Pairs approach is significantly better than the Event Phrase (EV) lookup approach at the $p < 0.1$ level.

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

Table 5: Analysis of Dictionary Combinations

Table 5 takes a closer look at how each pair of dictionaries performed. The first row shows that requiring a document to have an event phrase and a purpose phrase produces the best precision (100%) but with low recall (14%). The second row reveals that requiring a document to have an event phrase and an agent term yields better recall (47%) and high precision (94%). The third row shows that requiring a document to have a purpose phrase and an agent term produces the best recall (50%) but with slightly lower precision (85%). Finally, the last row of Table 5 shows that taking the union of these results (i.e., any combination of dictionary pairs is sufficient) yields the best recall (71%) with high precision (88%), demonstrating that we get the best coverage by recognizing multiple combinations of event information.

Lexicon	Recall	Precision	F-score
Seeds	13	87	22
Iter #1	50	88	63
Iter #2	63	89	74
Iter #3	68	88	77
Iter #4	71	88	79

Table 6: **All Pairs** Lookup Results using only Seeds and the Lexicons Learned after each Iteration, on the Test Set

Table 6 shows the performance of the lexicon lookup approach using the **All Pairs** criteria during the bootstrapping process. The first row shows the results using only 10 agent seeds and 4 purpose seeds as shown in Table 1. The following four rows in the table show the performance of **All Pairs** using

the lexicons learned after each bootstrapping iteration. We can see that the recall increases steadily and that precision is maintained at a high level throughout the bootstrapping process.

Event recognition can be formulated as an information retrieval (IR) problem. As another point of comparison, we ran an existing IR system, Terrier (Ounis et al., 2007), on our test set. We used Terrier to rank these 300 documents given our set of event keywords as the query [7], and then generated a recall/precision curve (Figure 4) by computing the precisions at different levels of recall, ranging from 0 to 1 in increments of *.10*. Terrier was run with the

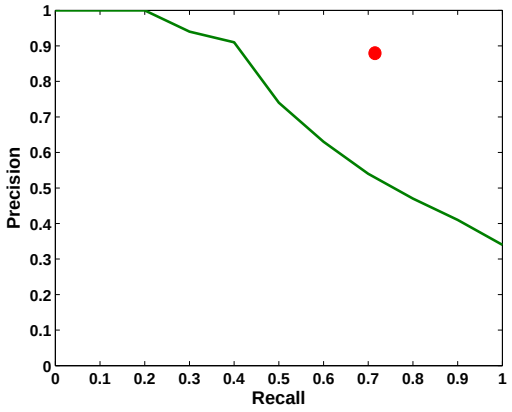

Figure 4: Comparison with the Terrier IR system

parameter PL2 which refers to an advanced Divergence From Randomness weighting model (Amati and Van Rijsbergen, 2002). In addition, Terrier used automatic query expansion. We can see that Terrier identified the first 60 documents (20% recall) with 100% precision. But precision dropped sharply after that. The circle in Figure 4 shows the performance of our bootstrapped dictionaries using the **All Pairs** approach. At comparable level of precision (88%), Terrier achieved about 45% recall versus 71% recall produced with the bootstrapped dictionaries.

4.4 Supervised Classifiers with Bootstrapped Dictionaries

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

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

Table 7: Supervised classifiers using the dictionaries

Table 7 shows the results for the five classifiers. **TermLex** encodes a binary feature for every phrase in any of our dictionaries. **PairLex** encodes a binary feature for each pair of phrases from two different dictionaries and requires them to occur in the same sentence. The TermLex classifier achieves good performance (74% F-score), but is not as effective as our All Pairs dictionary look-up approach (79% F-score). The PairLex classifier yield higher precision but very low recall, undoubtedly due to sparsity issues in matching specific pairs of phrases.

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

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

[7] We gave Terrier one query with all of the event keywords.

4.5 Finding Articles with no Event Keyword

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

We performed an additional set of experiments with documents in the Gigaword corpus that contain no human-selected civil unrest keyword. Following our multi-faceted approach to event recognition, we collected all documents that contain a sentence that matches phrases in at least two of our bootstrapped event dictionaries. This process retrieved 178,197 documents. The first column of Table 8 shows the number of documents that had phrases found in two different dictionaries (EV+AG, EV+PU, AG+PU) or in all three dictionaries (EV+AG+PU).

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

Table 8: Evaluation of articles with no event keyword

We randomly sampled 50 documents from each category and had them annotated. The accuracies are shown in the third column. Finding all three types of phrases produced the best accuracy, 74%. Furthermore, we found over 6,800 documents that had all three types of event information using our learned dictionaries. This result demonstrates that the bootstrapped dictionaries can recognize many event descriptions that would have been missed by searching only with manually selected keywords. This experiment also confirms that multi-facted event recognition using all three learned dictionaries achieves good accuracy even for documents that do not contain the civil unrest keywords.

5 Conclusions

We proposed a *multi-faceted* approach to event recognition and presented a bootstrapping technique to learn event phrases as well as agent terms and purpose phrases associated with civil unrest events. Our results showed that *multi-faceted event recognition* using the learned dictionaries achieved high accuracy and performed better than several other methods. The bootstrapping approach can be easily trained for new domains since it requires only a large collection of unannotated texts and a few event keywords, agent terms, and purpose phrases for the events of interest. Furthermore, although the training phase requires syntactic parsing to learn the event dictionaries, the dictionaries can then be used for event recognition without needing to parse the documents.

An open question for future work is to investigate whether the same multi-faceted approach to event recognition will work well for other types of events. Our belief is that many different types of events have characteristic agent terms, but additional types of facets will need to be defined to cover a broad array of event types. The syntactic constructions used to harvest dictionary items may also vary depending on the types of event information that must be learned. In future research, we plan to explore these issues in more depth to design a more general multi-faceted event recognition system, and we plan to investigate new ways to use these event dictionaries for event extraction as well.

6 Acknowledgments

This research was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI / NBC) contract number D12PC00285 and by the National Science Foundation under grant IIS-1018314. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBE, NSF, or the U.S. Government.

References

ACE Evaluations. 2006. *http://www.itl.nist.gov/iad/mig/tests/ace/*.

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

J. Allan, V. Lavrenko, and H. Jin. 2000a. First Story Detection in TDT is Hard. In *Proceedings of the 2000 ACM CIKM International Conference on Information and Knowledge Management*.

J. Allan, Victor Lavrenko, Daniella Malin, and Russell Swan. 2000b. Detections, Bounds, and Timelines: UMass and TDT-3. In *Proceedings of Topic Detection and Tracking Workshop*.

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

G. Amati and C. J. Van Rijsbergen. 2002. Probabilistic Models of Information Retrieval based on Measuring Divergence from Randomness. *ACM Transactions on Information Systems*, 20(4):357–389.

D. Appelt, J. Hobbs, J. Bear, D. Israel, and M. Tyson. 1993. FASTUS: a finite-state processor for information extraction from real-world text. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*.

H. Becker, M. Naaman, and L. Gravano. 2011. Beyond trending topics: Real-world event identification on twitter. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*.

E. Benson, A. Haghighi, and R. Barzilay. 2011. Event discovery in social media feeds.

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

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.

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*.

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

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.

R. Huang and E. Riloff. 2011. Peeling Back the Layers: Detecting Event Role Fillers in Secondary Contexts.

H. Jin, R. Schwartz, S. Sista, and F. Walls. 1999. Topic Tracking for Radio, TV broadcast, and Newswire. In *EUROSPEECH*.

T. Joachims. 1999. Making Large-Scale Support Vector Machine Learning Practical. In A. Smola B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA.

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

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

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

M. Mathioudakis and N. Koudas. 2010. TwitterMonitor: trend detection over the twitter stream. In *Proceedings of the 2010 international conference on Management of data*, page 11551158. ACM.

D. Metzler, C. Cai, and E. Hovy. 2012. Structured Event Retrieval over Microblog Archives. In *Proceedings of The 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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

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

R. Parker, D. Graff, J. Kong, K. Chen, and Kazuaki M. 2011. English Gigaword. In *Linguistic Data Consortium*.

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. Petrovic, M. Osborne, and V. Lavrenko. 2012. Using Paraphrases for Improving First Story Detection in

News and Twitter. In *Proceedings of The 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

A.-M. Popescu, M. Pennacchiotti, and D. A. Paranjpe. 2011. Extracting events and event descriptions from twitter.

E. Riloff and W. Lehnert. 1994. Information Extraction as a Basis for High-Precision Text Classification. *ACM Transactions on Information Systems*, 12(3):296–333, July.

E. Riloff and J. Lorenzen. 1999. Extraction-based text categorization: Generating domain-specific role relationships automatically. In Tomek Strzalkowski, editor, *Natural Language Information Retrieval*. Kluwer Academic Publishers.

E. Riloff. 1996. 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.

A. Ritter, Mausam, O. Etzioni, and S. Clark. 2012. Open domain event extraction from twitter. In *The Proceedings of The 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

T. Sakaki, M. Okazaki, and Y. Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors.

S. 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)*.

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)*.

A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. 2010. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*.

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)*.

X. Zhang, H. Fuehres, and P. A. Gloor. 2010. Predicting Stock Market Indicators Through Twitter ”I hope it is not as bad as I fear”. In *COINs*.