Automatically Extracting Nominal Mentions of Events with a Bootstrapped Probabilistic Classifier*

Cassandre Creswell[†] and **Matthew J. Beal**[‡] and **John Chen**[†]
Thomas L. Cornell[†] and **Lars Nilsson**[†] and **Rohini K. Srihari**[†‡]

[†]Janya, Inc.
1408 Sweet Home Road, Suite 1
Amherst NY 14228
{ccreswell,jchen,cornell,
lars,rohini}@janyainc.com

[‡]Dept. of Computer Science and Engineering
University at Buffalo
The State University of New York
Amherst NY 14260
mbeal@cse.buffalo.edu

Abstract

Most approaches to event extraction focus on mentions anchored in verbs. However, many mentions of events surface as noun phrases. Detecting them can increase the recall of event extraction and provide the foundation for detecting relations between events. This paper describes a weakly-supervised method for detecting nominal event mentions that combines techniques from word sense disambiguation (WSD) and lexical acquisition to create a classifier that labels noun phrases as denoting events or non-events. The classifier uses bootstrapped probabilistic generative models of the contexts of events and non-events. The contexts are the lexically-anchored semantic dependency relations that the NPs appear in. Our method dramatically improves with bootstrapping, and comfortably outperforms lexical lookup methods which are based on very much larger hand-crafted resources.

1 Introduction

The goal of information extraction is to generate a set of abstract information objects that represent the entities, events, and relations of particular types mentioned in unstructured text. For example, in a judicial domain, relevant event types might be ARREST, CHARGING, TRIAL, etc.

Although event extraction techniques usually focus on extracting mentions textually anchored by verb phrases or clauses, e.g. (Aone and Ramos-Santacruz, 2000), many event mentions, especially subsequent mentions of events that are the primary topic of a document, are referred to with nominals. Because of this, detecting nominal event mentions, like those in (1), can increase the recall of event extraction systems, in particular for the most important events in a document.[1]

(1) The slain journalist was a main organizer of **the massive demonstrations** that forced Syria to withdraw its troops from Lebanon last April, after Assad was widely accused of planning **Hariri's assassination** in **a February car bombing** that was similar to **today's blast**.

Detecting event nominals is also an important step in detecting relations between event mentions, as in the causal relation between the demonstrations and the withdrawal and the similarity relation between the bombing and the blast in (1).

Finally, detecting nominal events can improve detection and coreference of non-named mentions of non-event entities (e.g. persons, locations, and organizations) by removing event nominals from consideration as mentions of entities.

Current extraction techniques for verbally-anchored events rest on the assumption that most verb phrases denote eventualities. A system to extract untyped event mentions can output all constituents headed by a non-auxiliary verb with a filter to remove instances of *to be, to seem*, etc. A statistical or rule-based classifier designed to detect event mentions of specific types can then be applied to filter these remaining instances. Noun phrases, in contrast, can be used to denote anything—eventualities, entities, abstractions, and only some are suitable for event-type filtering.

* This work was supported in part by SBIR grant FA8750-05-C-0187 from the Air Force Research Laboratory (AFRL)/IFED.

[1] For example, in the 2005 Automatic Content Extraction training data, of the 5349 event mentions, over 35% (1934) were nominals.

1.1 Challenges of nominal event detection

Extraction of nominal mentions of events encompasses many of the fundamental challenges of natural language processing. Creating a general purpose lexicon of all potentially event-denoting terms in a language is a labor-intensive task. On top of this, even utilizing an existing lexical resource like WordNet requires sense disambiguation at run-time because event nominals display the full spectrum of sense distinction behaviors (Copestake and Briscoe, 1995), including idiosyncratic polysemy, as in (2); constructional polysemy, as in (3); coactivation, (4); and copredication, as in (5).

(2) a. On May 30 group of Iranian mountaineers hoisted the Iranian tricolor on **the summit**.

b. EU Leaders are arriving here for **their two-day summit** beginning Thursday.

(3) Things are getting back to normal in the Baywood Golf Club after **a chemical spill**[=event]. Clean-up crews said **the chemical spill**[=result] was 99 percent water and shouldn't cause harm to area residents.

(4) Managing partner Naimoli said he wasn't concerned about recent media **criticism**.

(5) The **construction** lasted 30 years and was inaugurated in the presence of the king in June 1684.

Given the breadth of lexical sense phenomena possible with event nominals, no existing approach can address all aspects. Lexical lookup, whether using a manually- or automatically-constructed resource, does not take context into consideration and so does not allow for vagueness or unknown words. Purely word-cooccurrence-based approaches (e.g. (Schütze, 1998)) are unsuitable for cases like (3) where both senses are possible in a single discourse. Furthermore, most WSD techniques, whether supervised or unsupervised, must be retrained for each individual lexical item, a computationally expensive procedure both at training and run time. To address these limitations, we have developed a technique which combines automatic lexical acquisition and sense disambiguation into a single-pass weakly-supervised algorithm for detecting event nominals.

The remainder of this paper is organized as follows: Section 2 describes our probabilistic classifier. Section 3 presents experimental results of this model, assesses its performance when bootstrapped to increase its coverage, and compares it to a lexical lookup technique. We describe related work in Section 4 and present conclusions and implications for future work in Section 5.

2 Weakly-supervised, simultaneous lexical acquisition and disambiguation

In this section we present a computational method that learns the distribution of context patterns that correlate with event vs. non-event mentions based on unambiguous seeds. Using these seeds we build two Bayesian probabilistic generative models of the data, one for non-event nominals and the other for event nominals. A classifier is then constructed by comparing the probability of a candidate instance under each model, with the winning model determining the classification. In Section 3 we show that this classifier's coverage can be increased beyond the initial labeled seed set by automatically selecting additional seeds from a very large unlabeled, parsed corpus.

The technique proceeds as follows. First, two lexicons of seed terms are created by hand. One lexicon includes nominal terms that are highly likely to unambiguously denote events; the other includes nominal terms that are highly likely to unambiguously denote anything other than events. Then, a very large corpus (>150K documents) is parsed using a broad-coverage dependency parser to extract all instantiations of a core set of semantic dependency relations, including verb-logical subject, verb-logical object, subject-nominal predicate, noun phrase-appositive-modifier, etc.

Format of data: Each instantiation is in the form of a dependency triple, (w_a, R, w_b), where R is the relation type and where each argument is represented just by its syntactic head, w_n. Each partial instantiation of the relation—i.e. either w_a or w_b is treated as a wild card $*$ that can be filled by any term—becomes a feature in the model. For every common noun term in the corpus that appears with at least one feature (including each entry in the seed lexicons), the times it appears with each feature are tabulated and stored in a matrix of counts. Each column of the matrix represents a feature, e.g. (*occur*,Verb-Subj, $*$); each row represents an individual term,[2] e.g. *murder*; and each entry is the number of times a term appeared with the feature in the corpus, i.e. as the instantiation of $*$. For each row, if the corresponding term appears in a lexicon it is given that designation, i.e. EVENT or NONEVENT, or if it does not appear in either lexicon, it is left unlabeled.

[2] A term is any common noun whether it is a single or multiword expression.

Probabilistic model: Here we present the details of the EVENT model—the computations for the NONEVENT model are identical. The probability model is built using a set of seeds words labeled as EVENTs and is designed to address two desiderata: **(I)** the EVENT model should assign high probability to an unlabeled vector, $\mathbf{v}$, if its features (as recorded in the count matrix) are similar to the vectors of the EVENT seeds; **(II)** each seed term s should contribute to the model in proportion to its prevalence in the training data.[3] These desiderata can be incorporated naturally into a mixture model formalism, where there are as many components in the mixture model as there are EVENT seed terms. Desideratum **(I)** is addressed by having each component of the mixture model assigning a multinomial probability to the vector, $\mathbf{v}$. For the ith mixture component built around the ith seed, $s^{(i)}$, the probability is

$$p(\mathbf{v}|\mathbf{s}^{(i)}) = \prod_{f=1}^{F} \left(\overline{s}_f^{(i)}\right)^{v_f} ,$$

where $\overline{s}_f^{(i)}$ is defined as the proportion of the times the seed was seen with feature f compared to the number of times the seed was seen with any feature $f' \in F$. Thus $\overline{s}_f^{(i)}$ is simply the (i, f)th entry in a row-sum normalized count matrix,

$$\overline{s}_f^{(i)} = \frac{s_f^{(i)}}{\sum_{f'=1}^{F} s_{f'}^{(i)}} .$$

Desideratum **(II)** is realized using a mixture density by forming a weighted mixture of the above multinomial distributions from all the provided seeds $i \in \mathcal{E}$. The weighting of the ith component is fixed to be the ratio of the number of occurrences of the ith EVENT seed, denoted $|\mathbf{s}^{(i)}|$, to the total number of all occurrences of event seed words. This gives more weight to more prevalent seed words:

$$p(\mathbf{s}^{(i)}) = \frac{|\mathbf{s}^{(i)}|}{\sum_{i' \in \mathcal{E}} |\mathbf{s}^{(i')}|} .$$

The EVENT generative probability is then:

$$p(\mathbf{v}|\text{EVENT}) = \sum_{i \in \mathcal{E}} \left[p(\mathbf{s}^{(i)}) \cdot p(\mathbf{v}|\mathbf{s}^{(i)}) \right] .$$

An example of the calculation for a model with just two event seeds and three features is given in Figure 1. A second model is built from the non-event seeds as well, and a corresponding probability $p(\mathbf{v}|\text{NONEVENT})$ is computed. The following difference (*log odds-ratio*)

$$d(\mathbf{v}) = \log p(\mathbf{v}|\text{EVENT}) - \log p(\mathbf{v}|\text{NONEVENT})$$

is then calculated. An instance v encoded as the vector $\mathbf{v}$ is labeled as EVENT or NONEVENT by examining the sign of $d(\mathbf{v})$. A positive difference $d(\mathbf{v})$ classifies v as EVENT; a negative value of $d(\mathbf{v})$ classifies v as NONEVENT. Should $d = 0$ the classifier is considered undecided and abstains.

		f_1	f_2	f_3
event seed vector	$\mathbf{s}^{(1)}$	3	1	8
event seed vector	$\mathbf{s}^{(2)}$	4	6	1
unlabeled mention vector	$\mathbf{v}$	2	0	7

$$p(\mathbf{v}|\text{event}) = \frac{12}{23} \cdot \left(\frac{3}{12}\right)^2 \left(\frac{1}{12}\right)^0 \left(\frac{8}{12}\right)^7 + \frac{11}{23} \cdot \left(\frac{4}{11}\right)^2 \left(\frac{6}{11}\right)^0 \left(\frac{1}{11}\right)^7 = 0.0019$$

Figure 1: Example of calculating the probability of unlabeled instance $\mathbf{v}$ under the event distribution composed of two event seeds $s^{(1)}$ and $s^{(2)}$.

Each test instance is composed of a term and the dependency triples it appears with in context in the test document. Therefore, an instance can be classified by **(i: *word*)**: Find the unlabeled feature vector in the training data corresponding to the term and apply the classifier to that vector, i.e. classify the instance based on the term's behavior summed across many occurrences in the training corpus; **(ii: *context*)**: Classify the instance based only on its immediate test context vector; or **(iii: *word+context*)**: For each model, multiply the probability information from the word vector (=i) and the context vector (=ii). In our experiments, all terms in the test corpus appeared at least once (80% appearing at least 500 times) in the training corpus, so there were no cases of unseen terms—not suprising with a training set 1800 times larger than the test set. However, the ability to label an instance based only on its immediate context means that there is a backoff method in the case of unseen terms.

3 Experimental Results

3.1 Training, test, and seed word data

In order to train and test the model, we created two corpora and a lexicon of event and non-event seeds. The training corpus consisted of 156,000 newswire documents, ∼100 million words, from the Foreign Broadcast Information Service, Lexis

[3] The counts used here are the number of times a term is seen with any feature in the training corpus because the indexing tool used to calculate counts does not keep track of which instances appeared simultaneously with more than one feature. We do not expect this artifact to dramatically change the relative seed frequencies in our model.

Nexis, and other online news archives. The corpus was parsed using Janya's information extraction application, Semantex, which creates both shallow, non-recursive parsing structures and dependency links, and all (w_i, R, w_j) statistics were extracted as described in Section 2. From the 1.9 million patterns, $(w_i, R, *)$ and $(*, R, w_j)$ extracted from the corpus, the 48,353 that appeared more than 300 times were retained as features.

The test corpus was composed of 77 additional documents ($\sim$56K words), overlapping in time and content but not included in the training set. These were annotated by hand to mark event nominals. Specifically, every referential noun phrase headed by a non-proper noun was considered for whether it denoted an achievement, accomplishment, activity, or process (Parsons, 1990). Noun heads denoting any of these were marked as EVENT, and all others were left unmarked.

All documents were first marked by a junior annotator, and then a non-blind second pass was performed by a senior annotator (first author). Several semantic classes were difficult to annotate because they are particularly prone to coactivation, including terms denoting financial acts, legal acts, speech acts, and economic processes. In addition, for terms like *mission, plan, duty, tactic, policy*, it can be unclear whether they are hyponyms of EVENT or another abstract concept. In every case, however, the mention was labeled as an event or non-event depending on whether its use in that context appeared to be more or less event-like, respectively. Tests for the "event-y"ness of the context included whether an unambiguous word would be an acceptable substitute there (e.g. *funds* [=only non-event] for *expenditure* [either]).

To create the test data, the annotated documents were also parsed to automatically extract all common noun-headed NPs and the dependency triples they instantiate. Those with heads that aligned with the offsets of an event annotation were labeled as events; the remainder were labeled as non-events. Because of parsing errors, about 10% of annotated event instances were lost, that is remained unlabeled or were labeled as non-events. So, our results are based on the set of recoverable event nominals as a subset of all common-noun headed NPs that were extracted. In the test corpus there were 9381 candidate instances, 1579 (17%) events and 7802 (83%) non-events. There were 2319 unique term types; of these, 167 types (7%) appeared both as event tokens and non-event tokens. Some sample ambiguous terms include: *behavior, attempt, settlement, deal, violation, progress, sermon, expenditure*.

We constructed two lexicons of nominals to use as the seed terms. For events, we created a list of **95** terms, such as *election, war, assassination, dismissal*, primarily based on introspection combined with some checks on individual terms in WordNet and other dictionaries and using Google searches to judge how "event-y" the term was.

To create a list of non-events, we used WordNet and the British National Corpus. First, from the set of all lexemes that appear in only one synset in WordNet, all nouns were extracted along with the topmost hypernym they appear under. From these we retained those that both appeared on a lemmatized frequency list of the 6,318 words with more than 800 occurrences in the whole 100M-word BNC (Kilgarriff, 1997) and had one of the hypernyms GROUP, PSYCHOLOGICAL, ENTITY, POSSESSION. We also retained select terms from the categories STATE and PHENOMENON were labeled non-event seeds. Examples of the **295** non-event seeds are *corpse, electronics, bureaucracy, airport, cattle*.

Of the 9381 test instances, 641 (6.8%) had a term that belonged to the seed list. With respect to types, 137 (5.9%) of the 2319 term types in the test data also appeared on the seed lists.

3.2 Experiments

Experiments were performed to investigate the performance of our models, both when using original seed lists, and also when varying the content of the seed lists using a bootstrapping technique that relies on the probabilistic framework of the model. A 1000-instance subset of the 9381 test data instances was used as a validation set; the remaining 8381 were used as evaluation data, on which we report all results (with the exception of Table 3 which is on the full test set).

EXP1: Results using original seed sets Probabilistic models for non-events and events were built from the full list of 295 non-event and 95 event seeds, respectively, as described above. Table 1 (top half: original seed set) shows the results over the 8381 evaluation data instances when using the three classification methods described above: (i) word, (ii) context, and (iii) word+context. The first row (ALL) reports scores where all undecided responses are marked as in-

correct. In the second row (FAIR), undecided answers ($d = 0$) are left out of the total, so the number of correct answers stays the same, but the percentage of correct answers increases.[4] Scores are measured in terms of accuracy on the EVENT instances, accuracy on the NONEVENT instances, TOTAL accuracy across all instances, and the simple AVERAGE of accuracies on non-events and events (last column). The AVERAGE score assumes that performance on non-events and events is equally important to us.

From EXP1, we see that the behavior of a term across an entire corpus is a better source of information about whether a particular instance of that term refers to an event than its immediate context. We can further infer that this is because the immediate context only provides definitive evidence for the models in 63.0% of cases; when the context model is not penalized for indecision, its accuracy improves considerably. Nonetheless, in combination with the word model, immediate context does not appear to provide much additional information over only the word. In other words, based only on a term's distribution in the past, one can make a reasonable prediction about how it will be used when it is seen again. Consequently, it seems that a well-constructed, i.e. domain customized, lexicon can classify nearly as well as a method that also takes context into account.

EXP2: Results on ACE 2005 event data In addition to using the data set created specifically for this project, we also used a subset of the annotated training data created for the ACE 2005 Event Detection and Recognition (VDR) task. Because only event mentions of specific types are marked in the ACE data, only recall of ACE event nominals can be measured rather than overall recall of event nominals and accuracy on non-event nominals. Results on the 1934 nominal mentions of events (omitting cases of $d = 0$) are shown in Table 2. The performance of the hand-crafted Lexicon 1 on the ACE data, described in Section 3.3 below, is also included.

The fact that our method performs somewhat better on the ACE data than on our own data, while the lexicon approach is worse (7 points higher vs. 3 points lower, respectively) can likely be explained by the fact that in creating our introspective seed set for events, we consulted the annotation manual for ACE event types and attempted to include in our list any unambiguous seed terms that fit those types.

[4]Note that Att(%) does not change with bootstrapping—an artifact of the sparsity of certain feature vectors in the training and test data, and not the model's constituents seeds.

Input Vector	Acc (%)	Att (%)
word	96.1	97.2
context	72.8	63.1
word+context	95.5	98.9
LEX 1	76.5	100.0

Table 2: **(EXP2)** Results on ACE event nominals: %correct (accuracy) and %attempted, for our classifiers and LEX 1.

EXP3: Increasing seed set via Bootstrapping There are over 2300 unlabeled vectors in the training data that correspond to the words that appear as lexical heads in the test data. These unlabeled training vectors can be powerfully leveraged using a simple bootstrapping algorithm to improve the individual models for non-events and events, as follows: **Step 1:** For each vector $\mathbf{v}$ in the unlabeled portion of training data, row-sum normalize it to produce $\tilde{\mathbf{v}}$ and compute a normalized measure of confidence of the algorithm's prediction, given by the magnitude of $d(\tilde{\mathbf{v}})$. **Step 2:** Add those vectors most confidently classified as either non-events or events to the seed set for non-events or events, according to the sign of $d(\tilde{\mathbf{v}})$. **Step 3:** Recalculate the model based on the new seed lists. **Step 4:** Repeat Steps 1–3 until either no more unlabeled vectors remain or the validation accuracy no longer increases.

In our experiments we added vectors to each model such that the ratio of the size of the seed sets remained constant, i.e. 50 non-events and 16 events were added at each iteration. Using our validation set, we determined that the bootstrapping should stop after 15 iterations (despite continuing for 21 iterations), at which point the average accuracy leveled out and then began to drop. After 15 iterations the seed set is of size $(295, 95) + (50, 16) \times 15 = (1045, 335)$. Figure 2 shows the change in the accuracy of the model as it is bootstrapped through 15 iterations.

TOTAL accuracy improves with bootstrapping, despite EVENT accuracy decreasing, because the test data is heavily populated with non-events, whose accuracy increases substantially. The AVERAGE accuracy also increases, which proves that bootstrapping is doing more than simply shifting the bias of the classifier to the majority class. The figure also shows that the final bootstrapped classifier comfortably outperforms Lexicon 1, impressive because the lexicon contains at least 13 times more terms than the seed lists.

			EVENT			NONEVENT			TOTAL			AVERAGE
		Input Vector	Correct	Acc (%)	Att (%)	Correct	Acc (%)	Att (%)	Correct	Acc (%)	Att (%)	Acc (%)
ORIGINAL SEED SET	ALL	word	1236	87.7	100.0	4217	60.5	100.0	5453	65.1	100.0	74.1
		context	627	44.5	100.0	2735	39.2	100.0	3362	40.1	100.0	41.9
		word+context	1251	**88.8**	100.0	4226	**60.6**	100.0	5477	**65.4**	100.0	**74.7**
	FAIR	word	1236	89.3	98.3	4217	60.7	99.6	5453	65.5	99.4	75.0
		context	627	69.4	64.2	2735	62.5	62.8	3362	63.6	63.0	65.9
		word+context	1251	**89.3**	99.5	4226	**60.7**	99.9	5477	**65.5**	99.8	**75.0**
BOOTSTRAPPED SEED SET	ALL	word	1110	78.8	100.0	5517	79.1	100.0	6627	79.1	100.0	79.0
		context	561	39.8	100.0	2975	42.7	100.0	3536	42.2	100.0	41.3
		word+context	1123	**79.8**	100.0	5539	**79.4**	100.0	6662	**79.5**	100.0	**79.6**
	FAIR	word	1110	80.2	98.3	5517	79.4	99.6	6627	79.5	99.4	79.8
		context	561	62.1	64.2	2975	67.9	62.8	3536	66.9	63.0	65.0
		word+context	1123	**80.2**	99.5	5539	**79.5**	99.9	6662	**79.7**	99.8	**79.9**
		LEX 1	1114	79.1	100.0	5074	72.8	100.0	6188	73.8	100.0	75.9
		total counts	1408			6973			8381			

Table 1: **(EXP1, EXP3)** Accuracies of classifiers in terms of correct classifications, % correct, and % attempted (if allowed to abstain), on the **evaluation test set**. **(Row 1)** Classifiers built from original seed set of size $(295, 95)$; **(Row 2)** Classifiers built from 15 iterations of bootstrapping; **(Row 3)** Classifier built from Lexicon 1. Accuracies in **bold** are those plotted in related Figures 2, 3(a) and 3(b).

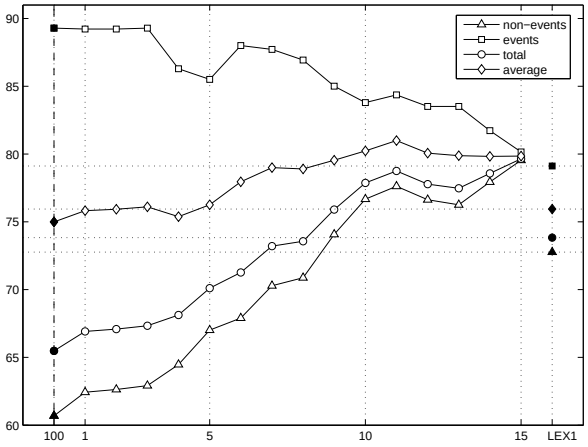


Figure 2: Accuracies vs. iterations of bootstrapping. Bold symbols on left denote classifier built from initial $(295, 95)$ seeds; and bold (disconnected) symbols at right are LEX 1.

EXP4: Bootstrapping with a reduced number of seeds The size of the original seed lists were chosen somewhat arbitrarily. In order to determine whether similar performance could be obtained using fewer seeds, i.e. less human effort, we experimented with reducing the size of the seed lexicons used to initialize the bootstrapping.

To do this, we randomly selected a fixed fraction, $f\%$, of the $(295, 95)$ available event and non-event seeds, and built a classifier from this subset of seeds (and discarded the remaining seeds). We then bootstrapped the classifier's models using the 4-step procedure described above, using candidate seed vectors from the unlabeled training corpus, and incrementing the number of seeds until the classifier consisted of $(295, 95)$ seeds. We then performed 15 additional bootstrapping iterations, each adding $(50, 16)$ seeds. Since the seeds making up the initial classifier are chosen stochastically, we repeated this entire process 10 times and report in Figures 3(a) and 3(b) the mean of the total and average accuracies for these 10 folds, respectively. Both plots have five traces, with each trace corresponding the fraction $f = (20, 40, 60, 80, 100)\%$ of labeled seeds used to build the initial models. As a point of reference, note that initializing with 100% of the seed lexicon corresponds to the first point of the traces in Figure 2 (where the x-axis is marked with $f = 100\%$).

Interestingly, there is no discernible difference in accuracy (total or average) for fractions f greater than 20%. However, upon bootstrapping we note the following trends. First, Figure 3(b) shows that using a larger initial seed set increases the maximum achievable accuracy, but this maximum occurs after a greater number bootstrapping iterations; indeed the maximum for 100% is achieved at 15 (or greater) iterations. This reflects the difference in rigidity of the initial models, with smaller initial models more easily misled by the seeds added by bootstrapping. Second, the final accuracies (total and average) are correlated with the initial seed set size, which is intuitively satisfying. Third, it appears from Figure 3(a) that the total accuracy at the model size (295,95) (or 100%) is in fact *anti*-correlated with the size of the initial seed set, with 20% performing best. This is correct, but highlights the sometimes misleading interpretation of the total accuracy: in this case the model is defaulting to classifying anything as a non-event (the majority class), and has a considerably impoverished event model.

If one wants to do as well as Lexicon 1 after 15 iterations of bootstrapping then one needs at least an initial seed set of size 60%. An alternative is

	EVENT		NONEVENT		TOTAL		AVERAGE
	Corr	(%)	Corr	(%)	Corr	(%)	(%)
LEX 1	1256	**79.5**	5695	**73.0**	6951	**74.1**	**76.3**
LEX 2	1502	95.1	4495	57.6	5997	63.9	76.4
LEX 3	349	22.1	7220	92.5	7569	80.7	57.3
Total	1579		7802		9381		

Table 3: Accuracy of several lexicons, showing number and percentage of correct classifications on the **full test set**.

to perform fewer iterations, but here we see that using 100% of the seeds comfortably achieves the highest total and average accuracies anyway.

3.3 Comparison with existing lexicons

In order to compare our weakly-supervised probabilistic method with a lexical lookup method based on very large hand-created lexical resources, we created three lexicons of event terms, which were used as very simple classifiers of the test data. If the test instance term belongs to the lexicon, it is labeled EVENT; otherwise, it is labeled as NONEVENT. The results on the full test set using these lexicons are shown in Table 3.

Lex 1 5,435 entries from NomLex (Macleod et al., 1998), FrameNet(Baker et al., 1998), CELEX (CEL, 1993), Timebank(Day et al., 2003).

Lex 2 13,659 entries from WordNet 2.0 hypernym classes EVENT, ACT, PROCESS, COGNITIVE PROCESS, & COMMUNICATION combined with Lex 1.

Lex 3 Combination of pre-existing lexicons in the information extraction application from WordNet, Oxford Advanced Learner's Dictionary, etc.

As shown in Tables 1 and 3, the relatively knowledge-poor method developed here using around 400 seeds performs well compared to the use of the much larger lexicons. For the task of detecting nominal events, using Lexicon 1 might be the quickest practical solution. In terms of extensibility to other semantic classes, domains, or languages lacking appropriate existing lexical resources, the advantage of our trainable method is clear. The primary requirement of this method is a dependency parser and a system user-developer who can provide a set of seeds for a class of interest and its complement. It should be possible in the next few years to create a dependency parser for a language with no existing linguistic resources (Klein and Manning, 2002). Rather than having to spend the considerable person-years it takes to create resources like FrameNet, CELEX, and WordNet, a better alternative will be to use weakly-supervised semantic labelers like the one described here.

4 Related Work

In recent years an array of new approaches have been developed using weakly-supervised techniques to train classifiers or learn lexical classes or synonyms, e.g. (Mihalcea, 2003; Riloff and Wiebe, 2003). Several approaches make use of dependency triples (Lin, 1998; Gorman and Curran, 2005). Our vector representation of the behavior of a word type across all its instances in a corpus is based on Lin (1998)'s DESCRIPTION OF A WORD.

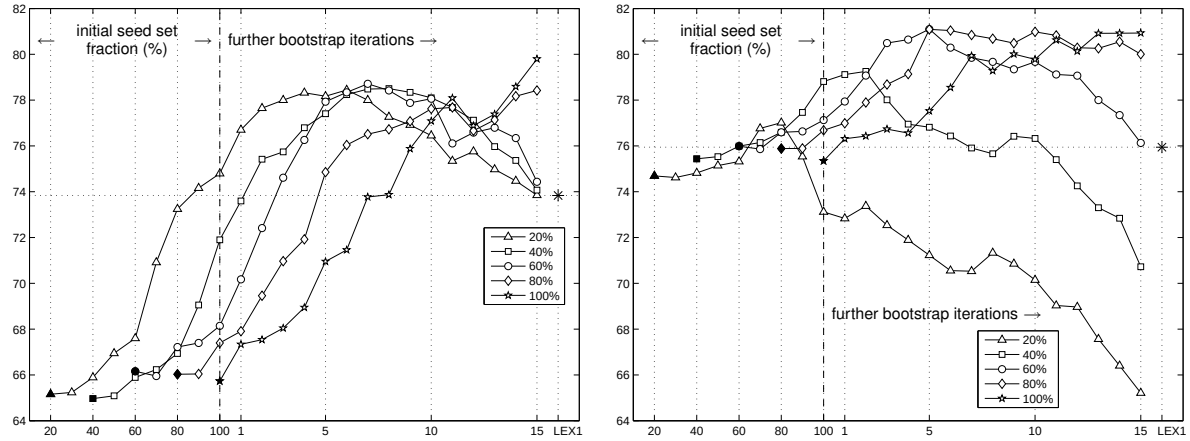
Yarowsky (1995) uses a conceptually similar technique for WSD that learns from a small set of seed examples and then increases recall by bootstrapping, evaluated on 12 idiosyncratically polysemous words. In that task, often a *single* disambiguating feature can be found in the context of a polysemous word instance, motivating his use of the decision list algorithm. In contrast, the goal here is to learn how event-like or non-event-like a *set* of contextual features together are. We do not expect that many individual features correlate unambiguously with references to events (or non-events), only that the presence of certain features make an event interpretation more or less likely. This justifies our probabilistic Bayesian approach, which performs well given its simplicity.

Thelen and Riloff (2002) use a bootstrapping algorithm to learn semantic lexicons of nouns for six semantic categories, one of which is EVENTS. For events, only 27% of the 1000 learned words are correct. Their experiments were on a much smaller scale, however, using the 1700 document MUC-4 data as a training corpus and using only 10 seeds per category.

Most prior work on event nominals does not try to classify them as events or non-events, but instead focuses on labeling the argument roles based on extrapolating information about the argument structure of the verbal root (Dahl et al., 1987; Lapata, 2002; Pradhan et al., 2004). Meyers, et al. (1998) describe how to extend a tool for extraction of verb-based events to corresponding nominalizations. Hull and Gomez (1996) design a set of rule-based algorithms to determine the sense of a nominalization and identify its arguments.

5 Conclusions

We have developed a novel algorithm for labeling nominals as events that combines WSD and lexical acquisition. After automatically bootstrapping the seed set, it performs better than static lexicons many times the original seed set size. Also, it is



(a) Total Accuracy (b) Average Accuracy

Figure 3: Accuracies of classifiers built from different-sized initial seed sets, and then bootstrapped onwards to the equivalent of 15 iterations as before. Total **(a)** and Average **(b)** accuracies highlight different aspects of the bootstrapping mechanism. Just as in Figure 2, the initial model is denoted with a bold symbol in the left part of the plot. Also for reference the relevant Lexicon 1 accuracy (LEX 1) is denoted with a $*$ at the far right.

a more robust than lexical lookup as it can also classify unknown words based on their immediate context and can remain agnostic in the absence of sufficient evidence.

Future directions for this work include applying it to other semantic labeling tasks and to domains other than general news. An important unresolved issue is the difficulty of formulating an appropriate seed set to give good coverage of the complement of the class to be labeled without the use of a resource like WordNet.

References

C. Aone and M. Ramos-Santacruz. 2000. REES: A large-scale relation and event extraction system. In *6th ANLP*, pages 79–83.

C. F. Baker, C. J. Fillmore, and J. B. Lowe. 1998. The Berkeley FrameNet project. In *Proc. COLING-ACL*.

Centre of Lexical Information, Nijmegen, 1993. *CELEX English database*, E25, online edition.

A. Copestake and T. Briscoe. 1995. Semi-productive polysemy and sense extension. *Journal of Semantics*, 12:15–67.

D. Dahl, M. Palmer, and R. Passonneau. 1987. Nominalizations in PUNDIT. In *Proc. of the 25th ACL*.

D. Day, L. Ferro, R. Gaizauskas, P. Hanks, M. Lazo, J. Pustejovsky, R. Sauri, A. See, A. Setzer, and B. Sundheim. 2003. The TIMEBANK corpus. In *Corpus Linguistics 2003*, Lancaster UK.

J. Gorman and J. Curran. 2005. Approximate searching for distributional similarity. In *Proc. of the ACL-SIGLEX Workshop on Deep Lexical Acquisition*, pages 97–104.

R. Hull and F. Gomez. 1996. Semantic interpretation of nominalizations. In *Proc. of the 13th National Conf. on Artificial Intelligence*, pages 1062–1068.

A. Kilgarriff. 1997. Putting frequencies in the dictionary. *Int'l J. of Lexicography*, 10(2):135–155.

D. Klein and C. Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proc. of the 40th ACL*.

M. Lapata. 2002. The disambiguation of nominalisations. *Computational Linguistics*, 28(3):357–388.

D. K. Lin. 1998. Automatic retrieval and clustering of similar words. In *Proc. of COLING-ACL '98*.

C. Macleod, R. Grishman, A. Meyers, L. Barrett, and R. Reeves. 1998. NOMLEX: A lexicon of nominalizations. In *Proc. of EURALEX'98*.

A. Meyers, C. Macleod, R. Yangarber, R. Grishman, L. Barrett, and R. Reeves. 1998. Using NOMLEX to produce nominalization patterns for information extraction. In *Proc. of the COLING-ACL Workshop on the Computational Treatment of Nominals*.

R. Mihalcea. 2003. Unsupervised natural language disambiguation using non-ambiguous words. In *Proc. of Recent Advances in Natural Language Processing*, pages 387–396.

T. Parsons. 1990. *Events in the Semantics of English*. MIT Press, Boston.

S. Pradhan, H. Sun, W. Ward, J. Martin, and D. Jurafsky. 2004. Parsing arguments of nominalizations in English and Chinese. In *Proc. of HLT-NAACL*.

E. Riloff and J. Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proc. EMNLP*.

H. Schütze. 1998. Automatic word sense disambiguation. *Computational Linguistics*, 24(1):97–124.

M. Thelen and E. Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proc. of EMNLP*.

D. Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. of the 33rd ACL*, pages 189–196.