Building a Semantic Lexicon of English Nouns via Bootstrapping

Ting Qian[1], **Benjamin Van Durme**[2] **and Lenhart Schubert**[2]
[1]Department of Brain and Cognitive Sciences
[2]Department of Computer Science
University of Rochester
Rochester, NY 14627 USA
ting.qian@rochester.edu, {vandurme, schubert}@cs.rochester.edu

Abstract

We describe the use of a weakly supervised bootstrapping algorithm in discovering contrasting semantic categories from a source lexicon with little training data. Our method primarily exploits the patterns in sentential contexts where different categories of words may appear. Experimental results are presented showing that such automatically categorized terms tend to agree with human judgements.

1 Introduction

There are important semantic distinctions between different types of English nouns. For example, some nouns typically refer to a concrete physical object, such as *book*, *tree*, etc. Others are used to represent the process or the result of an event (e.g. *birth*, *celebration*). Such information is useful in disambiguating syntactically similar phrases and sentences, so as to provide more accurate semantic interpretations. For instance, A MAN WITH HOBBIES and A MAN WITH APPLES share the same structure, but convey very different aspects about the *man* being referred to (i.e. activities vs possessions).

Compiling such a lexicon by hand, e.g., WordNet (Fellbaum, 1998), requires tremendous time and expertise. In addition, when new words appear, these will have to be analyzed and added manually. Furthermore, a single, global lexicon may contain erroneous categorizations when used within a specific domain/genre; we would like a "flexible" lexicon, adaptable to a given corpus. Also, in adapting semantic classifications of words to a particular genre or domain, we would like to be able to exploit continuing improvements in methods of extracting semantic occurrence patterns from text.

We present our initial efforts in discovering semantic classes incrementally under a weakly supervised bootstrapping process. The approach is able to selectively learn from its own discoveries, thereby minimizing the effort needed to provide seed examples as well as maintaining a reasonable accuracy rate. In what follows, we first focus on its application to an event-noun classification task, and then use a physical-object vs non-physical-object experiment as a showcase for the algorithm's generality.

2 Bootstrapping Algorithm

The bootstrapping algorithm discovers words with semantic properties similar to a small set of labelled seed examples. These examples can be manually selected from an existing lexicon. By simply changing the semantic property of the seed set, this algorithm can be applied to the task of discovering a variety of semantic classes.

Features Classification is performed using a perceptron-based model (Rosenblatt, 1958) that examines *features* of each word. We use two kinds of features in our model: morphological (affix and word length), and contextual. Suffixes, such as *-ion*, often reveal the semantic type that a noun belongs to (e.g., *destruction*, *explosion*). Other suffixes like *-er* typically suggest non-event nouns (e.g. *waiter*, *hanger*). The set of affixes can be modified to reflect meaningful distinctions in the task at hand. Regarding word length, longer words tend to have more

Proceedings of the NAACL HLT Student Research Workshop and Doctoral Consortium, pages 37–42,
Boulder, Colorado, June 2009.

syllables, and thus are more likely to contain affixes. For example, if a word ends with *-ment*, its number of letters must be ≥ 5. We defined a partition of words based on word length: shortest (fewer than 5 letters), short (5-7), medium (8-12), long (13-19), and longest (> 19).

Besides morphological features, we also make use of verbalized propositions resulting from the experiments of Van Durme et al. (2008) as contextual features. These outputs are in the form of world knowledge "factoids" abstracted from texts, based on logical forms from parsed sentences, produced by the KNEXT system (see Schubert (2002) for details). The followings are some sample factoids about the word *destruction*, extracted from the British National Corpus.

- A PERSON-OR-ORGANIZATION MAY UNDERGO A DESTRUCTION
- INDIVIDUAL -S MAY HAVE A DESTRUCTION
- PROPERTY MAY UNDERGO A DESTRUCTION

We take each verbalization (with the target word removed) as a contextual feature, such as `PROPERTY MAY UNDERGO A __`. Words from the same semantic category (e.g., event nouns) should have semantic and syntactic similarities on the sentential level. Thus their contextual features, which reflect the use of words both semantically and syntactically, should be similar. For instance, `PROPERTY MAY UNDERGO A PROTECTION` is another verbalization produced by KNEXT, suggesting the word *protection* may belong to the same category as *destruction*.

A few rough-and-ready heuristics are already employed by KNEXT to do the same task as we wish to automate here. A built-in classifier judges nominals to be event or non-event ones based on analysis of endings, plus a list of event nouns whose endings are unrevealing, and a list of non-event nouns whose endings tend to suggest they are event nouns. As a result, the factoids used as contextual features in our work already reflect the built-in classifier's attempt to distinguish event nouns from the rest. Thus, the use of these contextual features may bias the algorithm to perform seemingly well on event-noun classification. However, we will show that our algorithm works for classification of other semantic categories, for which KNEXT does not yet have discriminative procedures.

Iterative Training We use a bootstrapping procedure to iteratively train a perceptron-based linear classifier. A perceptron algorithm determines whether the active features of a test case are similar to those learned from given categories of examples. In an iterative training process, the classifier first learns from a small seed set, which contains examples of all categories (in binary classification, both positive and negative examples) manually selected to reflect human knowledge of semantic categories. The classifier then discovers new instances (and corresponding features) of each category. Based on activation values, these newly discovered instances are selectively admitted into the original training set, which increases the size of training examples for the next iteration.

The iterative training algorithm described above is adopted from Klementiev and Roth (2006). The advantage of bootstrapping is the ability to automatically learn from new discoveries, which saves both time and effort required to manually examine a source lexicon. However, if implemented exactly as described above, this process has two apparent disadvantages: New examples may be wrongly classified by the model; and it is difficult to evaluate the discriminative models produced in successive iterations, as there are no standard data against which to judge them (the new examples are by definition previously unexamined). We propose two measures to alleviate these problems. First, we admit into the training set only those instances whose activation values are higher than the mean activation of their corresponding categories in an iteration. This sets a variable threshold that is correlated with the performance of the model at each iteration. Second, we evaluate iterative results *post hoc*, using a *Bootstrapping Score*. This measures the efficacy of bootstrapping (i.e. the ratio of correct newly discovered instances to training examples) and precision (i.e. the proportion of correct discoveries among all those returned by the algorithm). We compute this score to decide which iteration has yielded the optimal discriminative model.

3 Building an Event-noun Lexicon

We applied the bootstrapping algorithm to the task of discovering event nouns from a source lexicon.

Event nouns are words that typically describe the occurrence, the process, or the result of an event. We first explore the effectiveness of this algorithm, and then describe a method of extracting the optimal model. Top-ranked features in the optimal model are used to find subcategories of event nouns.

Experimental Setup The WordNet noun-list is chosen as the source lexicon (Fellbaum, 1998), which consists of 21,512 nouns. The purpose of this task is to explore the separability of event nouns from this collection.

typical suffixes: *appeasement, arrival, renewal, construction, robbery, departure, happening*
irregular cases: *birth, collapse, crash, death, decline, demise, loss, murder*

Table 1: Examples of event-nouns in initial training set.

We manually selected 15 event nouns and 215 non-event nouns for the seed set. Event-noun examples are representative of subcategories within the semantic class, as well as their commonly seen morphological structures (Table 1). Non-event examples are primarily exceptions to morphological regularities (to prevent the algorithm from overly relying on affix features), such as, *anything*, *ambition*, *diagonal*. The subset of all contextual and morphological features represented by both event and non-event examples are used to bootstrap the training process.

Event Noun Discovery Reducing the number of working features is often an effective strategy in training a perceptron. We experimented with two cut-off thresholds for features: in Trial 1, features must appear at least 10 times (55,354 remaining); in Trial 2, features must appear at least 15 times (35,457 remaining).

We set the training process to run for 20 iterations in both trials. Classification results of each iteration were collected. We expect the algorithm to discover few event nouns during early iterations. But with new instances found in each subsequent iteration, it ought to utilize newly seen features and discover more. Figure 1 confirms our intuition.

The number of classified event-noun instances increased sharply at the 15th iteration in Trial 1 and the 11th iteration in Trial 2, which may suggest overfitting of the training examples used in those iterations. If so, this should also correlate with an increase of error rate in the classification results (error rate defined as the percentage of non-event nouns identified as event nouns in all discovered event nouns). We manually marked all misclassified event noun instances for the first 10 iterations in both trials. The error rate in Trial 2 is expected to significantly increase at the 10th iteration, while Trial 1 should exhibit little increase in error rate within this interval. This expectation is confirmed in Figure 2.

Extracting the Optimal Model We further pursued the task of finding the iteration that has yielded the best model. Optimality is judged from two aspects: 1) the number of correctly identified event nouns should be significantly larger than the size of seed examples; and 2) the accuracy of classification results should be relatively high so that it takes little effort to clean up the result. Once the optimal model is determined, we analyze its most heavily weighted features and try to derive finer categories from them. Furthermore, the optimal model could be used to discover new instances from other source lexicons in the future.

We define a measure called the Bootstrapping Score (BS), serving a similar purpose as an F-score. BS is computed as in Formula (1).

$$BS = \frac{2 * BR * Precision}{BR + Precision}\,. \qquad (1)$$

Here the Bootstrapping Rate (BR) is computed as:

$$BR = \frac{|NEW|}{|NEW| + |SEED|}\,, \qquad (2)$$

where $|NEW|$ is the number of correctly identified new instances (seed examples excluded), and $|SEED|$ is the size of seed examples. The rate of bootstrapping reveals how large the effect of the bootstrapping process is. Note that BR is different from the classic measure *recall*, for which the total number of relevent documents (i.e. true event nouns in English) must be known *a priori* – again, this knowledge is what we are discovering. The score is a *post hoc* solution; both BR and precision are computed for analysis after the algorithm has finished. Combining Formulas (1) and (2), a higher Bootstrapping Score means better model quality.

Bootstrapping scores of models in the first ten iterations are plotted in Figure 3. Model quality in

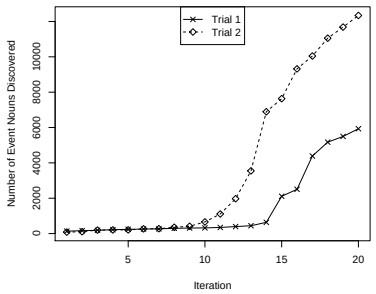

Figure 1: Classification rate

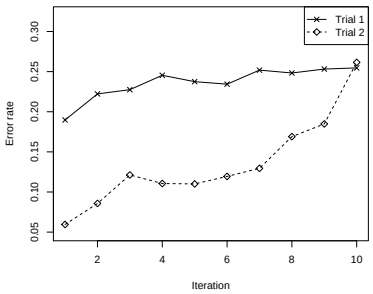

Figure 2: Error rate

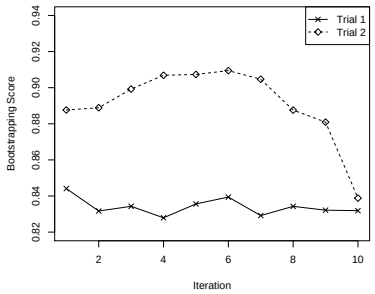

Figure 3: Bootstrapping score

	1	...	**6**	...	**10**
incorrect	5	...	32	...	176
correct	79	...	236	...	497
error rate	5.9%	...	11.9%	...	26.2%
score	87.0%	...	90.8%	...	83.8%

Table 2: From iterations 1 to 10, comparison between instance counts, error rates, and bootstrapping scores as the measure of model quality.

Trial 2 is better than in Trial 1 on average. In addition, within Trial 2, Iteration 6 yielded the best discriminative model with a bootstrapping score of 90.8%. Compared to instance counts and error rate measures as shown in Table 2, this bootstrapping score provides a balanced measure of model quality. The model at the 6th iteration (hereafter, Model 6) can be considered the optimal model generated during the bootstrapping training process.

Top-ranked Features in the Optimal Model In order to understand why Model 6 is optimal, we extracted its top 15 features that activate the event-noun target in Model 6, as listed in Table 3. Interestingly, top-ranked features are all contextual ones. In fact, in later models where the ranks of morphological features are boosted, the algorithm performed worse as a result of relying too much on those context-insensitive features.

Collectively, top-ranked features define the contextual patterns of event nouns. We are interested in finding semantic subcategories within the set of event nouns (497 nouns, Trial 2) by exploiting these features individually. For instance, some events typically happen to people only (e.g. *birth*, *betrayal*), while others usually happen to inanimate objects (e.g. *destruction*, *removal*). Human actions can also be distinguished by the number of participants, such as group activities (e.g. *election*) or individual activities (e.g. *death*). It is thus worth distinguishing nouns that describe different sorts of events.

Manual Classification We extracted the top 100 contextual features from Model 6 and grouped them into *feature classes*. A feature class consists of contextual features sharing similar meanings. For instance, `A COUNTRY MAY UNDERGO __` and `A STATE MAY UNDERGO __` both belong to the class *social activity*. For each feature class, we enumerate all words that correspond to its feature instances. Examples are shown in Table 4.

Not all events can be unambiguously classified into one of the subcategories. However, this is also not necessary because these categories overlap with one another. For example, *death* describes an event that tends to occur both individually and briefly. In addition to the six categories listed here, new categories can be added by creating more feature classes.

Automatic Clustering Representing each noun as a frequency vector over the top 100 most discriminating contextual features, we employed k-means clustering and compared the results to our manually crafted subcategories.

Through trial-and-error, we set k to 12, with the smallest resulting cluster containing 2 nouns (*interpretation* and *perception*), while the biggest resulting cluster contained 320 event nouns (that seemed to share no apparent semantic properties). Other clusters varied from 5 to 50 words in size, with examples shown in Table 5.

The advantage of automatic clustering is that the results may reflect an English speaker's impression of word similarity gained through language use. Un-

a person-or-organization may undergo a _	a state may undergo a _	a _ can be attempted
a country may undergo a _	a child may have a _	a _ can be for a country
a company may undergo a _	a project may undergo a _	authority may undergo a _
an explanation can be for a _	an empire may undergo a _	a war may undergo a _
days may have a _	a _ can be abrupt	a _ can be rapid

Table 3: Top 15 features that promote activation of the event-noun target, ranked from most weighted to least.

human events: adoption, arrival, birth, betrayal, death, development, disappearance, emancipation, funeral ...
events of inanimate objects: collapse, construction, definition, destruction, identification, inception, movement, recreation, removal ...
individual activities: birth, death, execution, funeral, promotion ...
social activities: abolition, evolution, federation, fragmentation, invasion ...
lasting events: campaign, development, growth, trial ...
brief events: awakening, collapse, death, mention, onset, resignation, thunderstorm ...

Table 4: Six subcategories of event nouns.

Cluster 1 (17): cancellation, cessation, closure, crackle, crash, demise, disappearance, dismissal, dissolution, division, introduction, onset, passing, resignation, reversal, termination, transformation
Cluster 2 (32): alienation, backing, betrayal, contemplation, election, execution, funeral, hallucination, imitation, juxtaposition, killing, mention, moulding, perfection, prosecution, recognition, refusal, removal, resurrection, semblance, inspection, occupation, promotion, trial ...
Cluster 3 (7): development, achievement, arrival, birth, death, loss, survival

Table 5: Examples resulting from automatic clustering.

fortunately, the discovered clusters do not typically come with the same obvious semantic properties as were defined in manual classification. In the example given above, neither of Cluster 1 and Cluster 3 seems to have a centralized semantic theme. But Cluster 2 seems to be mostly about human activities.

Comparison with WordNet To compare our results with WordNet resources, we enumerated all children of the gloss "*something that happens at a given place and time*", giving 7655 terms (phrases excluded). This gave a broader range of event nouns, such as proper nouns and procedures (e.g. *9/11*, *CT*, *MRI*), onomatopoeias (e.g. *mew*, *moo*), and words whose event reading is only secondary (e.g. *picture*, *politics*, *teamwork*). These types of words tend to have very different contextual features from what our algorithm had discovered.

While our method may be limited by the choice of seed examples, we were able to discover event nouns not classified under this set by WordNet, suggesting that the discovery mechanism itself is a robust one. Among them were low-frequency nouns (e.g. *crescendo*, *demise*, names of processes (e.g. *absorption*, *evolution*), and particular cases like *thunderstorm*.

4 Extension to Other Semantic Categories

To verify that our bootstrapping algorithm was not simply relying on KNEXT's own event classification heuristics, we set the algorithm to learn the distinction between physical and non-physical objects/entities.

(Non-)Physical-object Nouns 15 physical-object/entity nouns (e.g. *knife*, *ring*, *anthropologist*) and 34 non-physical ones (e.g. *happiness*, *knowledge*) were given to the model as the initial training set. At the 9th iteration, the number of discovered physical objects (which form the minority group between the two) approaches 2,200 and levels off. We randomly sampled five 20-word groups (a subset of these words are listed in Table 6) from this entire set of discovered physical objects, and computed an average error rate of 4%. Prominent features of the model at the 9th iteration are shown in Table 7.

5 Related Work

The method of using distributional patterns in a large text corpus to find semantically related En-

heifer, sheriff, collector, hippie, accountant, cape, scab, pebble, box, dick, calculator, sago, brow, ship, *?john*, superstar, border, rabbit, poker, garter, grinder, millionaire, ash, herdsman, *?cwm*, pug, bra, fulmar, **campaign*, stallion, deserter, boot, tear, elbow, cavalry, novel, cardigan, nutcase, *?bulge*, businessman, cop, fig, musician, spire, butcher, dog, elk, ...

Table 6: Physical-object nouns randomly sampled from results; words with an asterisk are misclassified, ones with a question mark are doubtful.

a male-individual can be a _	a _ can be small
a person can be a _	a _ can be large
a _ can be young	a _ can be german
-S**morphological feature*	a _ can be british
a _ can be old	a _ can be good

Table 7: Top-10 features that promote activation of the physical-object target in the model.

glish nouns first appeared in Hindle (1990). Roark and Charniak (1998) constructed a semantic lexicon using co-occurrence statistics of nouns within noun phrases. More recently, Liakata and Pulman (2008) induced a hierarchy over nominals using as features knowledge fragments similar to the sort given by KNEXT. Our work might be viewed as aiming for the same goal (a lexico-semantic based partitioning over nominals, tied to corpus-based knowledge), but allowing for an *a priori* bias regarding preferred structure.

The idea of bootstrapping lexical semantic properties goes back at least to Hearst (1998), where the idea is suggested of using seed examples of a relation to discover lexico-syntactic extraction patterns and then using these to discover further examples of the desired relation. The Basilisk system developed by Thelen and Riloff (2002) almost paralleled our effort. However, negative features – features that would prevent a word from being classified into a semantic category – were not considered in their model. In addition, in scoring candidate words, their algorithm only looked at the average relevance of syntactic patterns. Our perceptron-based algorithm examines the combinatorial effect of those patterns, which has yielded results suggesting improved accuracy and bootstrapping efficacy.

Similar to our experiments here using k-means, Lin and Pantel (2001) gave a clustering algorithm for iteratively building semantic classes, using as features argument positions within fragments from a syntactic dependency parser.

6 Conclusion

We have presented a bootstrapping approach for creating semantically tagged lexicons. The method can effectively classify nouns with contrasting semantic properties, even when the initial training set is a very small. Further classification is possible with both manual and automatic methods by utilizing individual contextual features in the optimal model.

Acknowledgments

This work was supported by NSF grants IIS-0328849 and IIS-0535105.

References

BNC Consortium. 2001. The British National Corpus, version 2 (BNC World). Distributed by Oxford University Computing Services.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Marti A. Hearst. 1998. Automated discovery of WordNet relations. In (Fellbaum, 1998), pages 131–153.

Donald Hindle. 1990. Noun classification from predicate-argument structures. In *ACL*.

Alexandre Klementiev and Dan Roth. 2006. Weakly supervised named entity transliteration and discovery from multilingual comparable corpora. In *ACL*.

Maria Liakata and Stephen Pulman. 2008. Automatic Fine-Grained Semantic Classification for Domain Adaption. In *Proceedings of Semantics in Text Processing (STEP)*.

Dekang Lin and Patrick Pantel. 2001. Induction of semantic classes from natural language text. In *KDD*.

Brian Roark and Eugene Charniak. 1998. Noun-phrase co-occurrence statistics for semi-automatic semantic lexicon construction. In *ACL*, pages 1110–1116.

Frank Rosenblatt. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

Lenhart K. Schubert. 2002. Can we derive general world knowledge from text? In *HLT*.

Michael Thelen and Ellen Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *EMNLP*.

Benjamin Van Durme, Ting Qian, and Lenhart Schubert. 2008. Class-driven Attribute Extraction. In *COLING*.