Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Philadelphia, July 2002, pp. 214-221. Association for Computational Linguistics.

A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Contexts

Michael Thelen and Ellen Riloff
School of Computing
University of Utah
Salt Lake City, UT 84112 USA
{thelenm,riloff}@cs.utah.edu

Abstract

This paper describes a bootstrapping algorithm called Basilisk that learns high-quality semantic lexicons for multiple categories. Basilisk begins with an unannotated corpus and seed words for each semantic category, which are then bootstrapped to learn new words for each category. Basilisk hypothesizes the semantic class of a word based on collective information over a large body of extraction pattern contexts. We evaluate Basilisk on six semantic categories. The semantic lexicons produced by Basilisk have higher precision than those produced by previous techniques, with several categories showing substantial improvement.

1 Introduction

In recent years, several algorithms have been developed to acquire semantic lexicons automatically or semi-automatically using corpus-based techniques. For our purposes, the term *semantic lexicon* will refer to a dictionary of words labeled with semantic classes (e.g., "bird" is an ANIMAL and "truck" is a VEHICLE). Semantic class information has proven to be useful for many natural language processing tasks, including information extraction (Riloff and Schmelzenbach, 1998; Soderland et al., 1995), anaphora resolution (Aone and Bennett, 1996), question answering (Moldovan et al., 1999; Hirschman et al., 1999), and prepositional phrase attachment (Brill and Resnik, 1994). Although some semantic dictionaries do exist (e.g., WordNet (Miller, 1990)), these resources often do not contain the specialized vocabulary and jargon that is needed for specific domains. Even for relatively general texts, such as the Wall Street Journal (Marcus et al., 1993) or terrorism articles (MUC-4 Proceedings, 1992), Roark and Charniak (Roark and Charniak, 1998) reported that 3 of every 5 terms generated by their semantic lexicon learner were not present in WordNet. These results suggest that automatic semantic lexicon acquisition could be used to enhance existing resources such as WordNet, or to produce semantic lexicons for specialized domains.

We have developed a weakly supervised bootstrapping algorithm called Basilisk that automatically generates semantic lexicons. Basilisk hypothesizes the semantic class of a word by gathering collective evidence about semantic associations from extraction pattern contexts. Basilisk also learns multiple semantic classes simultaneously, which helps constrain the bootstrapping process.

First, we present Basilisk's bootstrapping algorithm and explain how it differs from previous work on semantic lexicon induction. Second, we present empirical results showing that Basilisk outperforms a previous algorithm. Third, we explore the idea of learning multiple semantic categories simultaneously by adding this capability to Basilisk as well as another bootstrapping algorithm. Finally, we present results showing that learning multiple semantic categories simultaneously improves performance.

2 Bootstrapping using Collective Evidence from Extraction Patterns

Basilisk (Bootstrapping Approach to SemantIc Lexicon Induction using Semantic Knowledge) is a weakly supervised bootstrapping algorithm that automatically generates semantic lexicons. Figure 1 shows the high-level view of Basilisk's bootstrapping process. The input to Basilisk is an unannotated text corpus and a few manually defined *seed words* for each semantic category. Before bootstrapping begins, we run an extraction pattern learner over the corpus which generates patterns to extract every noun phrase in the corpus.

The bootstrapping process begins by selecting a subset of the extraction patterns that tend to extract the seed words. We call this the *pattern pool*.

The nouns extracted by these patterns become candidates for the lexicon and are placed in a *candidate word pool*. Basilisk scores each candidate word by gathering all patterns that extract it and measuring how strongly those contexts are associated with words that belong to the semantic category. The five best candidate words are added to the lexicon, and the process starts over again. In this section, we describe Basilisk's bootstrapping algorithm in more detail and discuss related work.

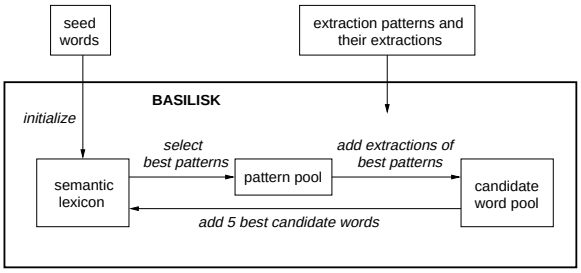

Figure 1: Basilisk Algorithm

2.1 Basilisk

The input to Basilisk is a text corpus and a set of seed words. We generated seed words by sorting the words in the corpus by frequency and manually identifying the 10 most frequent nouns that belong to each category. These seed words form the initial semantic lexicon. In this section we describe the learning process for a single semantic category. In Section 3 we will explain how the process is adapted to handle multiple categories simultaneously.

To identify new lexicon entries, Basilisk relies on extraction patterns to provide contextual evidence that a word belongs to a semantic class. As our representation for extraction patterns, we used the AutoSlog system (Riloff, 1996). AutoSlog's extraction patterns represent linguistic expressions that extract a noun phrase in one of three syntactic roles: subject, direct object, or prepositional phrase object. For example, three patterns that would extract people are: "*<subject> was arrested*", "*murdered <direct_object>*", and "*collaborated with <pp_object>*". Extraction patterns represent linguistic contexts that often reveal the meaning of a word by virtue of syntax and lexical semantics. Extraction patterns are typically designed to capture role relationships. For example, consider the verb "robbed" when it occurs in the active voice. The subject of "robbed" identifies the perpetrator, while the direct object of "robbed" identifies the victim or target.

Before bootstrapping begins, we run AutoSlog exhaustively over the corpus to generate an extraction pattern for every noun phrase that appears. The patterns are then applied to the corpus and all of their extracted noun phrases are recorded. Figure 2 shows the bootstrapping process that follows, which we explain in the following sections.

Generate all extraction patterns in the corpus and record their extractions.
lexicon = {seed words}
$i := 0$

BOOTSTRAPPING
1. Score all extraction patterns
2. *pattern_pool* = top ranked 20+i patterns
3. *candidate_word_pool* = extractions of patterns in *pattern_pool*
4. Score candidate words in *candidate_word_pool*
5. Add top 5 candidate words to *lexicon*
6. $i := i + 1$
7. Go to Step 1.

Figure 2: Basilisk's bootstrapping algorithm

2.1.1 The Pattern Pool and Candidate Pool

The first step in the bootstrapping process is to score the extraction patterns based on their tendency to extract known category members. All words that are currently defined in the semantic lexicon are considered to be category members. Basilisk scores each pattern using the *RlogF* metric that has been used for extraction pattern learning (Riloff, 1996). The score for each pattern is computed as:

$$RlogF(pattern_i) = \frac{F_i}{N_i} * \log_2(F_i) \qquad (1)$$

where F_i is the number of category members extracted by $pattern_i$ and N_i is the total number of nouns extracted by $pattern_i$. Intuitively, the *RlogF* metric is a weighted conditional probability; a pattern receives a high score if a high percentage of its extractions are category members, or if a moderate percentage of its extractions are category members and it extracts a lot of them.

The top N extraction patterns are put into a *pattern pool*. Basilisk uses a value of N=20 for the first iteration, which allows a variety of patterns to be considered, yet is small enough that all of the patterns are strongly associated with the category.[1]

[1] "Depleted" patterns are not included in this set. A pattern is depleted if all of its extracted nouns are already defined in the lexicon, in which case it has no unclassified words to contribute.

The purpose of the *pattern pool* is to narrow down the field of candidates for the lexicon. Basilisk collects all noun phrases (NPs) extracted by patterns in the *pattern pool* and puts the head noun of each NP into the *candidate word pool*. Only these nouns are considered for addition to the lexicon.

As the bootstrapping progresses, using the same value N=20 causes the candidate pool to become stagnant. For example, let's assume that Basilisk performs perfectly, adding only valid category words to the lexicon. After some number of iterations, all of the valid category members extracted by the top 20 patterns will have been added to the lexicon, leaving only non-category words left to consider. For this reason, the *pattern pool* needs to be infused with new patterns so that more nouns (extractions) become available for consideration. To achieve this effect, we increment the value of N by one after each bootstrapping iteration. This ensures that there is always at least one new pattern contributing words to the *candidate word pool* on each successive iteration.

2.1.2 Selecting Words for the Lexicon

The next step is to score the candidate words. For each word, Basilisk collects every pattern that extracted the word. All extraction patterns are used during this step, not just the patterns in the *pattern pool*. Initially, we used a scoring function that computes the average number of category members extracted by the patterns. The formula is:

$$score(word_i) = \frac{\sum_{j=1}^{P_i} F_j}{P_i} \quad (2)$$

where P_i is the number of patterns that extract $word_i$, and F_j is the number of distinct category members extracted by pattern j. A word receives a high score if it is extracted by patterns that also have a tendency to extract known category members.

As an example, suppose the word "Peru" is in the candidate word pool as a possible location. Basilisk finds all patterns that extract "Peru" and computes the average number of known locations extracted by those patterns. Let's assume that the three patterns shown below extract "Peru" and that the underlined words are known locations. "Peru" would receive a score of (2+3+2)/3 = 2.3. Intuitively, this means that patterns that extract "Peru" also extract, on average, 2.3 known location words.

"was killed in <np>"
Extractions: *Peru, clashes, a shootout, El Salvador, Colombia*

"<np> was divided"
Extractions: *the country, the Medellin cartel, Colombia, Peru, the army, Nicaragua*

"ambassador to <np>"
Extractions: *Nicaragua, Peru, the UN, Panama*

Unfortunately, this scoring function has a problem. The average can be heavily skewed by one pattern that extracts a large number of category members. For example, suppose word w is extracted by 10 patterns, 9 which do not extract any category members but the tenth extracts 50 category members. The average number of category members extracted by these patterns will be 5. This is misleading because the only evidence linking word w with the semantic category is a single, high-frequency extraction pattern (which may extract words that belong to other categories as well).

To alleviate this problem, we modified the scoring function to compute the average *logarithm* of the number of category members extracted by each pattern. The logarithm reduces the influence of any single pattern. We will refer to this scoring metric as the *AvgLog* function, which is defined below. Since $log_2(1) = 0$, we add one to each frequency count so that patterns which extract a single category member contribute a positive value.

$$AvgLog(word_i) = \frac{\sum_{j=1}^{P_i} \log_2(F_j + 1)}{P_i} \quad (3)$$

Using this scoring metric, all words in the *candidate word pool* are scored and the top five words are added to the semantic lexicon. The *pattern pool* and the *candidate word pool* are then emptied, and the bootstrapping process starts over again.

2.1.3 Related Work

Several weakly supervised learning algorithms have previously been developed to generate semantic lexicons from text corpora. Riloff and Shepherd (Riloff and Shepherd, 1997) developed a bootstrapping algorithm that exploits lexical co-occurrence statistics, and Roark and Charniak (Roark and Charniak, 1998) refined this algorithm to focus more explicitly on certain syntactic structures. Hale, Ge, and Charniak (Ge et al., 1998) devised a technique to learn the gender of words. Caraballo (Caraballo, 1999) and Hearst (Hearst, 1992) created techniques to learn hypernym/hyponym relationships. None of these previous algorithms used extraction patterns or similar contexts to infer semantic class associations.

Several learning algorithms have also been developed for named entity recognition (e.g., (Collins

and Singer, 1999; Cucerzan and Yarowsky, 1999)). (Collins and Singer, 1999) used contextual information of a different sort than we do. Furthermore, our research aims to learn general nouns (e.g., "artist") rather than proper nouns, so many of the features commonly used to great advantage for named entity recognition (e.g., capitalization and title words) are not applicable to our task.

The algorithm most closely related to Basilisk is meta-bootstrapping (Riloff and Jones, 1999), which also uses extraction pattern contexts for semantic lexicon induction. Meta-bootstrapping identifies a single extraction pattern that is highly correlated with a semantic category and then assumes that all of its extracted noun phrases belong to the same category. However, this assumption is often violated, which allows incorrect terms to enter the lexicon. Riloff and Jones acknowledged this issue and used a second level of bootstrapping (the "Meta" bootstrapping level) to alleviate this problem. While meta-bootstrapping trusts individual extraction patterns to make unilateral decisions, Basilisk gathers collective evidence from a large set of extraction patterns. As we will demonstrate in Section 2.2, Basilisk's approach produces better results than meta-bootstrapping and is also considerably more efficient because it uses only a single bootstrapping loop (meta-bootstrapping uses nested bootstrapping). However, meta-bootstrapping produces category-specific extraction patterns in addition to a semantic lexicon, while Basilisk focuses exclusively on semantic lexicon induction.

2.2 Single Category Results

To evaluate Basilisk's performance, we ran experiments with the MUC-4 corpus (MUC-4 Proceedings, 1992), which contains 1700 texts associated with terrorism. We used Basilisk to learn semantic lexicons for six semantic categories: BUILDING, EVENT, HUMAN, LOCATION, TIME, and WEAPON. Before we ran these experiments, one of the authors manually labeled every head noun in the corpus that was found by an extraction pattern. These manual annotations were the gold standard. Table 1 shows the breakdown of semantic categories for the head nouns. These numbers represent a baseline: an algorithm that randomly selects words would be expected to get accuracies consistent with these numbers.

Three semantic lexicon learners have previously been evaluated on the MUC-4 corpus (Riloff and Shepherd, 1997; Roark and Charniak, 1998; Riloff and Jones, 1999), and of these *meta-bootstrapping* achieved the best results. So we implemented the meta-bootstrapping algorithm ourselves to directly compare its performance with that of Basilisk. A difference between the original implementation and ours is that our version learns individual nouns (as does Basilisk) instead of noun phrases. We believe that learning individual nouns is a more conservative approach because noun phrases often overlap (e.g., "high-power bombs" and "incendiary bombs" would count as two different lexicon entries in the original meta-bootstrapping algorithm). Consequently, our meta-bootstrapping results differ from those reported in (Riloff and Jones, 1999).

Category	Total	Percentage
building	188	2.2%
event	501	5.9%
human	1856	21.9%
location	1018	12.0%
time	112	1.3%
weapon	147	1.7%
other	4638	54.8%

Table 1: Breakdown of semantic categories

Figure 3 shows the results for Basilisk (ba-1) and meta-bootstrapping (mb-1). We ran both algorithms for 200 iterations, so that 1000 words were added to the lexicon (5 words per iteration). The X axis shows the number of words learned, and the Y axis shows how many were correct. The Y axes have different ranges because some categories are more prolific than others. Basilisk outperforms meta-bootstrapping for every category, often substantially. For the human and location categories, Basilisk learned hundreds of words, with accuracies in the 80-89% range through much of the bootstrapping. It is worth noting that Basilisk's performance held up well on the human and location categories even at the end, achieving 79.5% (795/1000) accuracy for humans and 53.2% (532/1000) accuracy for locations.

3 Learning Multiple Semantic Categories Simultaneously

We also explored the idea of bootstrapping multiple semantic classes simultaneously. Our hypothesis was that errors of confusion[2] between semantic categories can be lessened by using information about multiple categories. This hypothesis makes sense only if a word cannot belong to more than one semantic class. In general, this is not true because words are often polysemous. But within a limited domain, a word usually has a dominant word sense. Therefore we make a "one sense per domain" assumption (similar

[2]We use the term *confusion* to refer to errors where a word is labeled as category X when it really belongs to category Y.

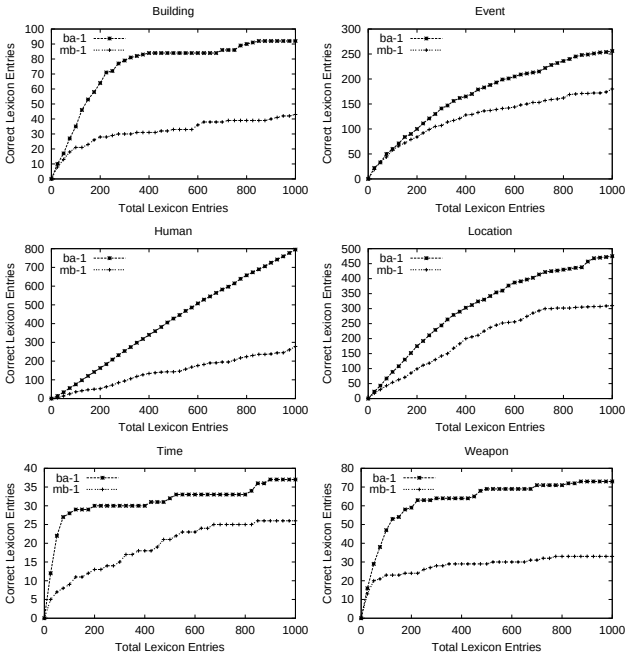

Figure 3: Basilisk and Meta-Bootstrapping Results, Single Category

to the "one sense per discourse" observation (Gale et al., 1992)) that a word belongs to a single semantic category within a limited domain. All of our experiments involve the MUC-4 terrorism domain and corpus, for which this assumption seems appropriate.

3.1 Motivation

Figure 4 shows one way of viewing the task of semantic lexicon induction. The set of all words in the corpus is visualized as a search space. Each category owns a certain territory within the space (demarcated with a dashed line), representing the words that are true members of that category. Not all territories are the same size, since some categories have more members than others.

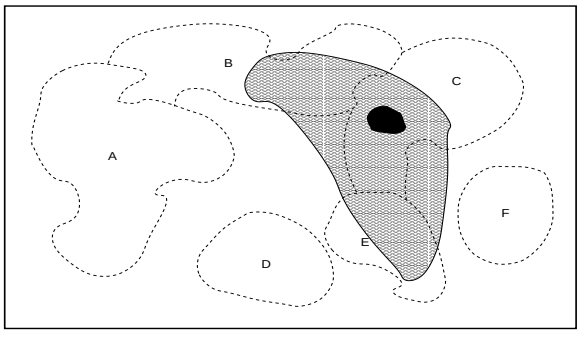

Figure 4: Bootstrapping a Single Category

Figure 4 illustrates what happens when a semantic lexicon is generated for a single category. The seed words for the category (in this case, category C) are represented by the solid black area in category C's territory. The hypothesized words in the growing lexicon are represented by a shaded area. The goal of the bootstrapping algorithm is to expand the area of hypothesized words so that it exactly matches the category's true territory. If the shaded area expands beyond the category's true territory, then incorrect words have been added to the lexicon. In Figure 4, category C has claimed a significant number of words that belong to categories B and E. When generating a lexicon for one category at a time, these confusion errors are impossible to detect because the learner has no knowledge of the other categories.

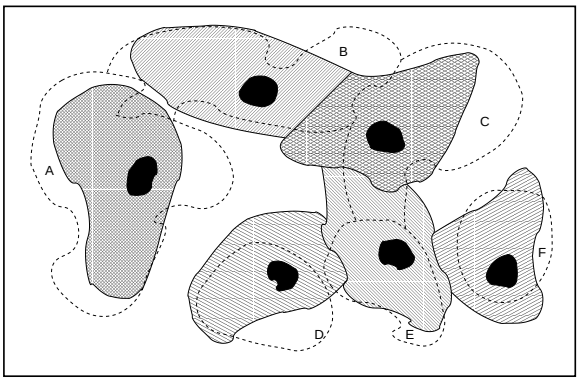

Figure 5: Bootstrapping Multiple Categories

Figure 5 shows the same search space when lexicons are generated for six categories simultaneously. If the lexicons cannot overlap, then we constrain the ability of a category to overstep its bounds. Category C is stopped when it begins to encroach upon the territories of categories B and E because words in those areas have already been claimed.

3.2 Simple Conflict Resolution

The easiest way to take advantage of multiple categories is to add simple conflict resolution that enforces the "one sense per domain" constraint. If more than one category tries to claim a word, then we use conflict resolution to decide which category should win. We incorporated a simple conflict resolution procedure into Basilisk, as well as the meta-bootstrapping algorithm. For both algorithms, the conflict resolution procedure works as follows. (1) If a word is hypothesized for category A but has already been assigned to category B during a previous iteration, then the category A hypothesis is discarded. (2) If a word is hypothesized for both category A and category B during the same iteration, then it

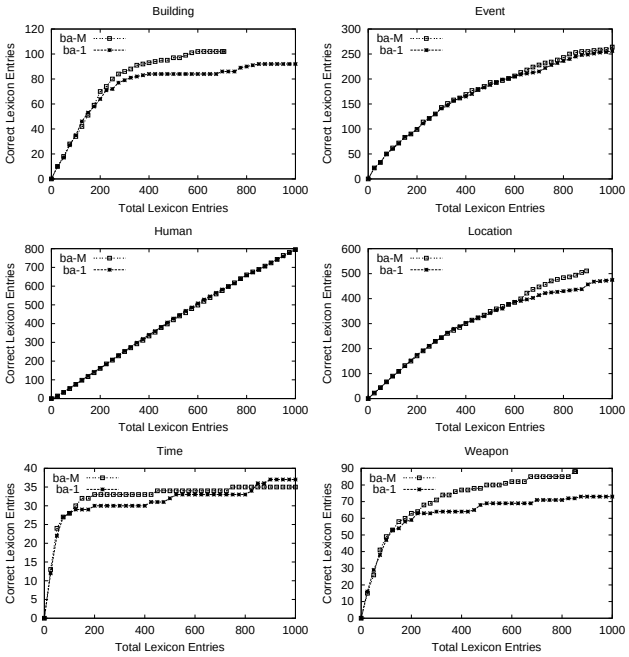

Figure 6: Basilisk, MCAT vs. 1CAT

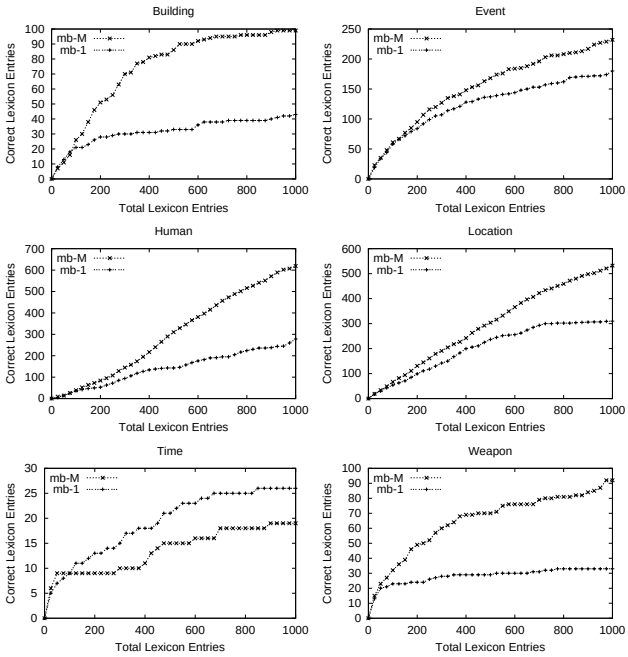

Figure 7: Meta-Bootstrapping, MCAT vs. 1CAT

is assigned to the category for which it receives the highest score. In Section 3.4, we will present empirical results showing how this simple conflict resolution scheme affects performance.

3.3 A Smarter Scoring Function for Multiple Categories

Simple conflict resolution helps the algorithm recognize when it has encroached on another category's territory, but it does not actively steer the bootstrapping in a more promising direction. A more intelligent way to handle multiple categories is to incorporate knowledge about other categories directly into the scoring function. We modified Basilisk's scoring function to prefer words that have strong evidence for one category but little or no evidence for competing categories. Each word w_i in the *candidate word pool* receives a score for category c_a based on the following formula:

$$\mathit{diff}(w_i,c_a) = \text{AvgLog}(w_i,c_a) - \max_{b \neq a} (\text{AvgLog}(w_i,c_b))$$

where *AvgLog* is the candidate scoring function used previously by Basilisk (see Equation 3) and the max function returns the maximum *AvgLog* value over all competing categories. For example, the score for each candidate LOCATION word will be its *AvgLog* score for the LOCATION category minus its maximum *AvgLog* score for all other categories. A word is ranked highly only if it has a high score for the targeted category and there is little evidence that it belongs to a different category. This has the effect of steering the bootstrapping process away from ambiguous parts of the search space.

3.4 Multiple Category Results

We will use the abbreviation 1CAT to indicate that only one semantic category was bootstrapped, and MCAT to indicate that multiple semantic categories were simultaneously bootstrapped. Figure 6 compares the performance of Basilisk-MCAT with conflict resolution (ba-M) against Basilisk-1CAT (ba-1). Most categories show small performance gains, with the BUILDING, LOCATION, and WEAPON categories benefitting the most. However, the improvement usually doesn't kick in until many bootstrapping iterations have passed. This phenomenon is consistent with the visualization of the search space in Figure 5. Since the seed words for each category are not generally located near each other in the search space, the bootstrapping process is unaffected by conflict resolution until the categories begin to encroach on each other's territories.

Figure 7 compares the performance of Meta-Bootstrapping-MCAT with conflict resolution (mb-M) against Meta-Bootstrapping-1CAT (mb-1). Learning multiple categories improves the performance of meta-bootstrapping dramatically for most categories. We were surprised that the improvement for meta-bootstrapping was much

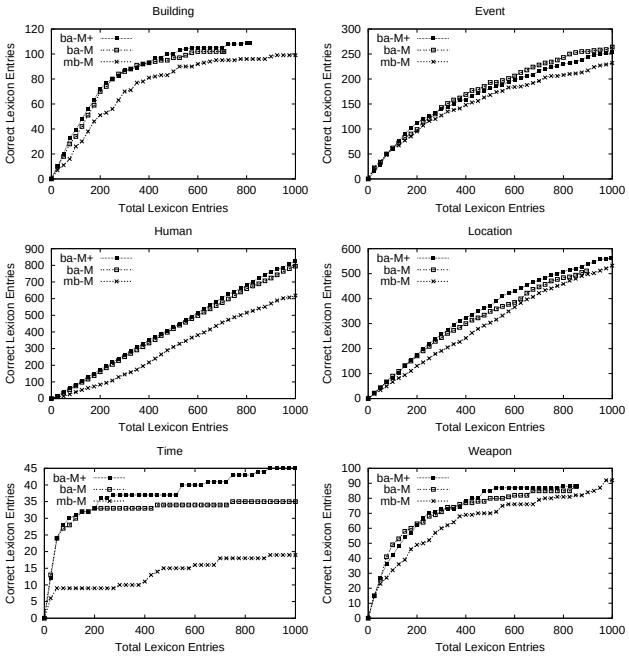

Figure 8: MetaBoot-MCAT vs. Basilisk-MCAT vs. Basilisk-MCAT+

more pronounced than for Basilisk. It seems that Basilisk was already doing a better job with errors of confusion, so meta-bootstrapping had more room for improvement.

Finally, we evaluated Basilisk using the *diff* scoring function to handle multiple categories. Figure 8 compares all three MCAT algorithms, with the smarter *diff* version of Basilisk labeled as ba-M+. Overall, this version of Basilisk performs best, showing a small improvement over the version with simple conflict resolution. Both multiple category versions of Basilisk also consistently outperform the multiple category version of meta-bootstrapping.

Table 2 summarizes the improvement of the best version of Basilisk over the original meta-bootstrapping algorithm. The left-hand column represents the number of words learned and each cell indicates how many of those words were correct. These results show that Basilisk produces substantially better accuracy and coverage than meta-bootstrapping.

Figure 9 shows examples of words learned by Basilisk. Inspection of the lexicons reveals many unusual words that could be easily overlooked by someone building a dictionary by hand. For example, the words "deserter" and "narcoterrorists" appear in a variety of terrorism articles but they are not commonly used words in general.

We also measured the recall of Basilisk's lexicons after 1000 words had been learned, based on the gold standard data shown in Table 1. The recall results range from 40-60%, which indicates that a good percentage of the category words are being found, although there are clearly more category words lurking in the corpus.

Total Words	MetaBoot 1CAT	Basilisk MCAT+
BUILDING		
100	21 (21.0%)	39 (39.0%)
200	28 (14.0%)	72 (36.0%)
500	33 (6.6%)	100 (20.0%)
800	39 (4.9%)	109 (13.6%)
1000	43 (4.3%)	n/a
EVENT		
100	61 (61.0%)	61 (61.0%)
200	89 (44.5%)	114 (57.0%)
500	146 (29.2%)	186 (37.2%)
800	172 (21.5%)	240 (30.0%)
1000	190 (19.0%)	266 (26.6%)
HUMAN		
100	36 (36.0%)	84 (84.0%)
200	53 (26.5%)	173 (86.5%)
500	143 (28.6%)	431 (86.2%)
800	224 (28.0%)	681 (85.1%)
1000	278 (27.8%)	829 (82.9%)
LOCATION		
100	54 (54.0%)	84 (84.0%)
200	99 (49.5%)	175 (87.5%)
500	237 (47.4%)	371 (74.2%)
800	302 (37.8%)	509 (63.6%)
1000	310 (31.0%)	n/a
TIME		
100	9 (9.0%)	30 (30.0%)
200	13 (6.5%)	33 (16.5%)
500	21 (4.2%)	37 (7.4%)
800	25 (3.1%)	43 (5.4%)
1000	26 (2.6%)	45 (4.5%)
WEAPON		
100	23 (23.0%)	42 (42.0%)
200	24 (12.0%)	62 (31.0%)
500	29 (5.8%)	85 (17.0%)
800	33 (4.1%)	88 (11.0%)
1000	33 (3.3%)	n/a

Table 2: Lexicon Results

4 Conclusions

Basilisk's bootstrapping algorithm exploits two ideas: (1) collective evidence from extraction patterns can be used to infer semantic category associations, and (2) learning multiple semantic categories simultaneously can help constrain the bootstrapping process. The accuracy achieved by Basilisk is substantially higher than that of previous techniques for semantic lexicon induction on the MUC-4 corpus, and empirical results show that both of Basilisk's ideas contribute to its performance. We also demon-

Building: theatre store cathedral temple palace penitentiary academy houses school mansions
Event: ambush assassination uprisings sabotage takeover incursion kidnappings clash shoot-out
Human: boys snipers detainees commandoes extremists deserter narcoterrorists demonstrators cronies missionaries
Location: suburb Soyapango capital Oslo regions cities neighborhoods Quito corregimiento
Time: afternoon evening decade hour March weeks Saturday eve anniversary Wednesday
Weapon: cannon grenade launchers firebomb car-bomb rifle pistol machineguns firearms

Figure 9: Example Semantic Lexicon Entries

strated that learning multiple semantic categories simultaneously improves the meta-bootstrapping algorithm, which suggests that this is a general observation which may improve other bootstrapping algorithms as well.

5 Acknowledgments

This research was supported by the National Science Foundation under award IRI-9704240.

References

Chinatsu Aone and Scott William Bennett. 1996. Applying machine learning to anaphora resolution. In *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Understanding*, pages 302–314. Springer-Verlag, Berlin.

E. Brill and P. Resnik. 1994. A Transformation-based Approach to Prepositional Phrase Attachment Disambiguation. In *Proceedings of the Fifteenth International Conference on Computational Linguistics (COLING-94)*.

S. Caraballo. 1999. Automatic Acquisition of a Hypernym-Labeled Noun Hierarchy from Text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 120–126.

M. Collins and Y. Singer. 1999. Unsupervised Models for Named Entity Classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Me thods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*.

S. Cucerzan and D. Yarowsky. 1999. Language Independent Named Entity Recognition Combining Morph ological and Contextual Evidence. In *Proceedings of the Joint SIGDAT Conference on Empirical Me thods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-99)*.

W. Gale, K. Church, and David Yarowsky. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26:415–439.

Niyu Ge, John Hale, and Eugene Charniak. 1998. A statistical approach to anaphora resolution. In *Proceedings of the Sixth Workshop on Very Large Corpora*.

M. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING-92)*.

Lynette Hirschman, Marc Light, Eric Breck, and John D. Burger. 1999. Deep Read: A reading comprehension system. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

George Miller. 1990. Wordnet: An on-line lexical database. In *International Journal of Lexicography*.

Dan Moldovan, Sanda Harabagiu, Marius Paşca, Rada Mihalcea, Richard Goodrum, Roxana Gîrju, and Vasile Rus. 1999. LASSO: A tool for surfing the answer net. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*.

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

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

E. Riloff and M. Schmelzenbach. 1998. An Empirical Approach to Conceptual Case Frame Acquisition. In *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 49–56.

E. Riloff and J. Shepherd. 1997. A Corpus-Based Approach for Building Semantic Lexicons. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 117–124.

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.

B. Roark and E. Charniak. 1998. Noun-phrase Co-occurrence Statistics for Semi-automatic Semantic Lexicon Construction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 1110–1116.

Stephen Soderland, David Fisher, Jonathan Aseltine, and Wendy Lehnert. 1995. CRYSTAL: Inducing a conceptual dictionary. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1314–1319.