To appear in *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*

Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping

Ellen Riloff
Department of Computer Science
University of Utah
Salt Lake City, UT 84112
riloff@cs.utah.edu

Rosie Jones
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
rosie@cs.cmu.edu

Abstract

Information extraction systems usually require two dictionaries: a semantic lexicon and a dictionary of extraction patterns for the domain. We present a multi-level bootstrapping algorithm that generates both the semantic lexicon and extraction patterns simultaneously. As input, our technique requires only unannotated training texts and a handful of seed words for a category. We use a *mutual bootstrapping* technique to alternately select the best extraction pattern for the category and bootstrap its extractions into the semantic lexicon, which is the basis for selecting the next extraction pattern. To make this approach more robust, we add a second level of bootstrapping (*meta-bootstrapping*) that retains only the most reliable lexicon entries produced by mutual bootstrapping and then restarts the process. We evaluated this multi-level bootstrapping technique on a collection of corporate web pages and a corpus of terrorism news articles. The algorithm produced high-quality dictionaries for several semantic categories.

Introduction

The purpose of *information extraction* (IE) systems is to extract domain-specific information from natural language text. IE systems typically rely on two domain-specific resources: a dictionary of extraction patterns and a semantic lexicon. The extraction patterns may be constructed by hand or may be generated automatically using one of several techniques. Most systems that generate extraction patterns automatically use special training resources, such as texts annotated with domain-specific tags (e.g., AutoSlog (Riloff 1993; 1996a), CRYSTAL (Soderland *et al.* 1995), RAPIER (Califf 1998), SRV (Freitag 1998), WHISK (Soderland 1999)) or manually defined keywords, frames, or object recognizers (e.g., PALKA (Kim & Moldovan 1993) and LIEP (Huffman 1996)). AutoSlog-TS (Riloff 1996b) takes a different approach by using a preclassified training corpus in which texts only need to be labeled as relevant or irrelevant to the domain. Semantic lexicons[1] for information extraction are almost always constructed by hand because general-purpose resources, such as WordNet (Miller 1990), do not contain the necessary domain-specific vocabulary. However there have been recent efforts to automate the construction of domain-specific semantic lexicons as well (Riloff & Shepherd 1997; Roark & Charniak 1998).

We explore the idea of learning both a dictionary of extraction patterns and a domain-specific semantic lexicon simultaneously. Furthermore, our technique requires no special training resources. The input to our algorithm is a set of unannotated training texts and a handful of "seed" words for the semantic category of interest. The heart of our approach is a *mutual bootstrapping* technique that learns extraction patterns from the seed words and then exploits the learned extraction patterns to identify more words that belong to the semantic category. We also introduce a second level of bootstrapping that retains only the most reliable lexicon entries from the results of mutual bootstrapping and restarts the process with the enhanced semantic lexicon. This two-tiered bootstrapping process is less sensitive to noise than a single level of bootstrapping and produces highly-quality dictionaries.

In this paper, we first describe the mutual bootstrapping algorithm that generates both a semantic lexicon and extraction patterns simultaneously. In the second section, we describe how the mutual bootstrapping process is itself bootstrapped to produce more accurate dictionaries at each iteration. In the third section, we present the results from experiments with two text collections: a set of corporate web pages, and a corpus of terrorism newswire articles.

Mutual Bootstrapping

Information extraction (IE) systems are designed to extract specific types of information from text. The categories of interest are defined in advance and usually require the extraction of noun phrases (NPs), such as the names of people, companies, or locations. For

[1]For our purposes, a semantic lexicon just refers to a dictionary of words with semantic category labels.

some IE tasks, the set of possible extractions is finite. For example, extracting country names from text is straightforward because it is easy to define a list of all countries. However, most IE tasks require the extraction of a potentially open-ended set of phrases. For example, it is impossible to enumerate all noun phrases that might describe a person, company, or location.

Most IE systems use both a semantic lexicon of known phrases and a dictionary of extraction patterns to recognize relevant noun phrases. For example, an IE system to identify locations might use a semantic lexicon that lists country names and the 50 U.S. states, and then rely on extraction patterns to recognize other location phrases such as cities, neighborhoods, and general descriptions like "downtown" or "northwest region". The semantic lexicon can also support the use of semantic constraints in the extraction patterns.

Our goal is to automate the construction of both a lexicon and extraction patterns for a semantic category using bootstrapping. The heart of our approach is based on the observation that extraction patterns can generate new examples of a semantic category, which in turn can be used to identify new extraction patterns. We will refer to this process as *mutual bootstrapping*.

The mutual bootstrapping process begins with a text corpus and a handful of predefined *seed words* for a semantic category. Before bootstrapping begins, the text corpus is used to generate a set of candidate extraction patterns. We used AutoSlog (Riloff 1993; 1996a) in an exhaustive fashion to generate extraction patterns for every noun phrase in the corpus. Given a noun phrase to extract, AutoSlog uses heuristics to generate a linguistic expression that represents relevant context for extracting the NP. This linguistic expression should be general enough to extract other relevant noun phrases as well. Because we applied AutoSlog exhaustively, the complete set of extraction patterns that it produced is capable of extracting every noun phrase in the training corpus. We then applied the extraction patterns to the corpus and recorded their extractions.

Using this data, the mutual bootstrapping procedure identifies the extraction pattern that is most useful for extracting known category members. This extraction pattern is then used to propose new phrases that belong in the semantic lexicon. Figure 1 outlines the mutual bootstrapping algorithm. At each iteration, the algorithm saves the best extraction pattern for the category to a list (*Cat_EPlist*). All of its extractions are assumed to be category members and are added to the semantic lexicon (*SemLex*). Then the next best extraction pattern is identified, based on both the original seed words plus the new words that were just added to the lexicon, and the process repeats. Since the semantic lexicon is constantly growing, the extraction patterns need to be rescored after each iteration. An important question is how long to run the bootstrapping loop. The simplest approach is to use a threshold cutoff, but we will discuss this issue more in the evaluation section.

The scoring heuristic is based on how many different lexicon entries a pattern extracts. This scoring metric rewards generality; a pattern that extracts a variety of category members will be scored higher than a pattern that extracts only one or two different category members, no matter how often. Scoring is also based on a "head phrase" matching scheme instead of requiring an exact match. Head phrase matching means that X matches Y if X is the rightmost substring of Y. For example, "New Zealand" will match any phrase that ends with "New Zealand", such as "eastern New Zealand" or "the modern day New Zealand". It would not match "the New Zealand coast" or just "Zealand". Head phrase matching is important for generality because any noun phrase can be preceded by an arbitrary number of modifiers.

Generate all candidate extraction patterns from the training corpus using AutoSlog.

Apply the candidate extraction patterns to the training corpus and save the patterns with their extractions to *EPdata*

$SemLex = \{\text{seed\_words}\}$

$Cat\_EPlist = \{\}$

MUTUAL BOOTSTRAPPING LOOP

1. Score all extraction patterns in *EPdata*.
2. *best_EP* = the highest scoring extraction pattern not already in *Cat_EPlist*
3. Add *best_EP* to *Cat_EPlist*
4. Add *best_EP*'s extractions to *SemLex*.
5. Go to step 1

Figure 1: Mutual Bootstrapping Algorithm

Each NP was stripped of leading articles, common adjectives (e.g., "his", "its", "other"), and numbers before being matched and saved to the lexicon. We used a small stopword list[2] and a number recognizer to discard overly general words such as pronouns and numbers. Using these criteria, we scored each extraction pattern using the *RlogF* metric used previously by AutoSlog-TS (Riloff 1996b). The score for extraction pattern i is computed as:

$$score(pattern_i) = R_i * log_2(F_i)$$

where F_i is the number of unique lexicon entries among the extractions produced by $pattern_i$, N_i is the total number of unique NPs that $pattern_i$ extracted, and $R_i = \frac{F_i}{N_i}$. This metric was designed for information extraction tasks, where it is important to identify not

[2]Most information retrieval systems use a *stopword list* to prevent extremely common words from being used for retrieval purposes. Our stopword list contained 35 words, mainly pronouns, determiners, and quantifiers.

only the most reliable extraction patterns but also patterns that will frequently extract relevant information (even if irrelevant information will also be extracted). For example, the pattern "kidnapped in <x>" will extract locations but it will also extract many dates (e.g., "kidnapped in January"). Even if it extracts dates and locations equally often, the fact that it frequently extracts locations makes it essential to have in the dictionary or many locations will be missed. Intuitively, the *RlogF* metric tries to strike a balance between reliability and frequency. The R value is high when the pattern's extractions are highly correlated with the semantic category, and the F value is high when the pattern extracts a large number of category members.

Figure 2 shows the results of the first five iterations of mutual bootstrapping to build location dictionaries from a terrorism corpus. Ten seed words were used: *bolivia*, *city*, *colombia*, *district*, *guatemala*, *honduras*, *neighborhood*, *nicaragua*, *region*, *town*. An asterisk after a noun phrase means that the noun phrase was acquired as a category member through bootstrapping. The F and N values for each pattern are shown in parentheses. Note that because of head phrase matching, "chapare region" will match the seed word "region" and be counted as a location when scoring the extraction pattern. But since that exact phrase was not in the lexicon before, it is considered to be a new location and added to it.

Best pattern	"headquartered in <x>" (F=3,N=4)
Known locations	*nicaragua*
New locations	*san miguel*, *chapare region*, *san miguel city*
Best pattern	"gripped <x>" (F=2,N=2)
Known locations	*colombia*, *guatemala*
New locations	none
Best pattern	"downed in <x>" (F=3,N=6)
Known locations	*nicaragua*, *san miguel**, *city*
New locations	*area*, *usulutan region*, *soyapango*
Best pattern	"to occupy <x>" (F=4,N=6)
Known locations	*nicaragua*, *town*
New locations	*small country*, *this northern area*, *san sebastian neighborhood*, *private property*
Best pattern	"shot in <x>" (F=5,N=12)
Known locations	*city*, *soyapango**
New locations	*jauja*, *central square*, *head*, *clash*, *back*, *central mountain region*, *air*, *villa el_salvador district*, *northwestern guatemala*, *left side*

Figure 2: Five iterations of mutual bootstrapping

Figure 2 shows both the strengths and weaknesses of the mutual bootstrapping approach. The extraction patterns are indicative of locations and have identified several new location phrases (e.g., *jauja*, *san miguel*, *soyapango*, and *this northern area*). But several non-location phrases have also been generated (e.g., *private property*, *head*, *clash*, *back*, *air*, *left side*). Most of these mistakes came from the pattern "shot in <x>", because this expression can refer to non-location phrases such as body parts. Also, most of these extraction patterns occur infrequently in the corpus. Although "headquartered in <x>" and "gripped <x>" are good location extractors, together they appeared only seven times in the 1500 training texts. As we will show in the next section, there are many other location patterns that occur much more frequently and are therefore more important to have in the dictionary.

Multi-level Bootstrapping

The mutual bootstrapping algorithm works well but its performance can deteriorate rapidly when non-category words enter the semantic lexicon. Once an extraction pattern is chosen for the dictionary, all of its extractions are immediately added to the lexicon and a few bad entries can quickly infect the dictionary. For example, if a pattern extracts dates as well as locations, then the dates are added to the lexicon and subsequent patterns are rewarded for extracting them.

To make the algorithm more robust, we introduced a second level of bootstrapping. The outer bootstrapping mechanism, which we call *meta-bootstrapping*, compiles the results from the inner (mutual) bootstrapping process and identifies the five most reliable lexicon entries. These five NPs are retained for the permanent semantic lexicon and the rest of the mutual bootstrapping process is discarded. The entire mutual bootstrapping process is then restarted from scratch. The meta-bootstrapping process is illustrated in Figure 3.

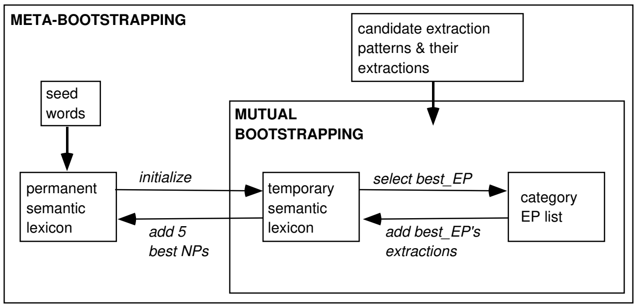

Figure 3: The Meta-Bootstrapping Process

To determine which NPs are most "reliable", we score each NP based on the number of different category patterns (members of *Cat_EPlist*) that extracted it. This criteria is based on the intuition that a noun phrase extracted by three different category patterns is more likely to belong to the category than a noun phrase extracted by only one pattern. We also add in a small factor to account for the strength of the patterns that extracted it. This is used mainly for tie-breaking purposes. The scoring formula is shown below, where N_i is the number of different category patterns that extracted NP_i.

Web Location Patterns	*Web Title Patterns*	*Web Company Patterns*	*Terrorism Location Patterns*	*Terrorism Weapon Patterns*
offices in <x>	served as <x>	owned by <x>	living in <x>	<x> exploded
facilities in <x>	became <x>	both as <x>	traveled to <x>	threw <x>
operations in <x>	company as <x>	<x> employed	become in <x>	bringing <x>
loans in <x>	to become <x>	<x> is distributor	sought in <x>	seized <x>
operates in <x>	experience as <x>	<x> positioning	presidents of <x>	quantity of <x>
locations in <x>	q. is <x>	marks of <x>	parts of <x>	surrender <x>
producer in <x>	appointed as <x>	motivated <x>	to enter <x>	search for <x>
states of <x>	to serve as <x>	<x> trust company	condemned in <x>	rocket <x>
seminars in <x>	elects <x>	sold to <x>	relations between <x>	<x> parked
activities in <x>	<x> capitalize	devoted to <x>	ministers of <x>	hurled <x>
consulting in <x>	williams is <x>	<x> consolidated stmts.	part in <x>	clips for <x>
countries of <x>	position of <x>	<x> thrive	taken in <x>	defused <x>
rep. of <x>	retired <x>	message to <x>	returned to <x>	million in <x>
outlets in <x>	expectations of <x>	<x> is obligations	process in <x>	confiscated <x>
consulting in <x>	promotion to <x>	<x> request information	involvement in <x>	<x> was hurled
customers in <x>	founded as <x>	<x> is foundation	intervention in <x>	placed <x>
diensten in <x>	established as <x>	<x> has positions	linked in <x>	rounds for <x>
distributors in <x>	assistant to <x>	incorporated as <x>	operates in <x>	consisted of <x>
services in <x>	meyerson is <x>	offices of <x>	kidnapped in <x>	firing <x>
expanded into <x>	<x> seated	<x> required to meet	refuge in <x>	explosion of <x>

Figure 4: Top 20 extraction patterns for 4 categories

$$score(NP_i) = \sum_{k=1}^{N_i} 1 + (.01 * score(pattern_k))$$

The main advantage of meta-bootstrapping comes from re-evaluating the extraction patterns after each mutual bootstrapping process. For example, after the first mutual bootstrapping run, five new words are added to the permanent semantic lexicon. Then mutual bootstrapping is restarted from scratch with the original seed words plus these five new words. Now, the best pattern selected by mutual bootstrapping might be different from the best pattern selected last time. This produces a snowball effect because its extractions are added to the temporary semantic lexicon which is the basis for choosing the next extraction pattern. In practice, what happens is that the ordering of the patterns changes (sometimes dramatically) between subsequent runs of mutual bootstrapping. In particular, more general patterns seem to float to the top as the permanent semantic lexicon grows.

Figure 4 shows the top 20 extraction patterns produced for several categories after 50 iterations of meta-bootstrapping. Note that the top five terrorism location patterns are different from the top five terrorism location patterns generated by mutual bootstrapping alone (shown in Figure 2). The top five patterns produced by meta-bootstrapping are much more common, extracting a total of 79 unique NPs, while the top five patterns produced by mutual bootstrapping extracted only 30 unique NPs.

Evaluation

To evaluate the meta-bootstrapping algorithm, we performed experiments with two text collections: corporate web pages collected for the WebKB project (Craven *et al.* 1998) and terrorism news articles from the MUC-4 corpus (MUC-4 Proceedings 1992). For training, we used 4160 of the web pages and 1500 of the terrorism texts. We preprocessed the web pages first by removing html tags and adding periods to separate independent phrases.[3] AutoSlog generated 19,690 candidate extraction patterns from the web page training set, and 14,064 candidate extraction patterns from the terrorism training set.[4] Then we ran the meta-bootstrapping algorithm on three semantic categories for the web pages (locations, person titles, and companies), and two semantic categories for the terrorism articles (locations and weapons). We used the seed word lists shown in Figure 5. We used different location seeds for the two text collections because the terrorism articles were mainly from Latin America while the web pages were much more international.

We ran the meta-bootstrapping algorithm (outer bootstrapping) for 50 iterations. The extraction patterns produced by the last iteration were the output of the system, along with the permanent semantic lexicon. For each meta-bootstrapping iteration, we ran the mutual bootstrapping procedure (inner bootstrapping) until it produced 10 patterns that extracted at least one new NP (i.e., not currently in the semantic

[3]Web pages pose a problem for NLP systems because separate lines do not always end with a period (e.g., list items and headers). We used several heuristics to insert periods whenever an independent line or phrase was suspected.

[4]AutoSlog actually generated many more extraction patterns, but for practical reasons we only used the patterns that appeared with frequency ≥ 2.

	Iter 1	*Iter 10*	*Iter 20*	*Iter 30*	*Iter 40*	*Iter 50*
Web Company	5/5 (1)	25/32 (.78)	52/65 (.80)	72/113 (.64)	86/163 (.53)	95/206 (.46)
Web Location	5/5 (1)	46/50 (.92)	88/100 (.88)	129/150 (.86)	163/200 (.82)	191/250 (.76)
Web Title	0/1 (0)	22/31 (.71)	63/81 (.78)	86/131 (.66)	101/181 (.56)	107/231 (.46)
Terr. Location	5/5 (1)	32/50 (.64)	66/100 (.66)	100/150 (.67)	127/200 (.64)	158/250 (.63)
Terr. Weapon	4/4 (1)	31/44 (.70)	68/94 (.72)	85/144 (.59)	101/194 (.52)	124/244 (.51)

Table 1: Accuracy of the Semantic Lexicons

Web Company:	*co. company corp. corporation inc. incorporated limited ltd. plc*
Web Location:	*australia canada china england france germany japan mexico switzerland united_states*
Web Title:	*ceo cfo president vice-president vp*
Terr. Location:	*bolivia city colombia district guatemala honduras neighborhood nicaragua region town*
Terr. Weapon:	*bomb bombs dynamite explosive explosives gun guns rifle rifles tnt*

Figure 5: Seed Word Lists

lexicon). But there were two exceptions: (1) if the best pattern had score < 0.7 then mutual bootstrapping stopped, or (2) if the best pattern had score > 1.8 then mutual bootstrapping continued. Intuitively, mutual bootstrapping stops when the best pattern looks especially dubious (its extractions would be risky to add to the lexicon) or keeps going if it is still generating strong extraction patterns. This scheme allows mutual bootstrapping to produce a variable number of extraction patterns, depending on how reliable it believes them to be. These criteria worked well empirically on the categories that we tested, but a more formal strategy is a worthwhile avenue for future research.

First, we evaluated the semantic lexicons in isolation by manually inspecting each word. We judged a word to belong to the category if it was a specific category member (e.g., "IBM" is a specific company) or a general referent for the category (e.g., "the company" is a referent for companies). Although referents are meaningless in isolation, they are useful for information extraction tasks because a coreference resolver should be able to find their antecedent. The referents were also very useful during bootstrapping because a pattern that extracts "the company" will probably also extract specific company names.

Table 1 shows the accuracy of the semantic lexicon after the 1st iteration of meta-bootstrapping and after each 10th iteration. Each cell shows the number of true category members among the entries generated thus far. For example, 32 phrases were added to company semantic lexicon after the tenth iteration and 25 of those (78%) were true company phrases. Table 1 shows that our algorithm found about 100-200 new phrases for all of the categories, and the density of good phrases was high. To put our results in perspective, other researchers have generated a semantic lexicon for the terrorism weapon category and achieved accuracy rates of 34/200 (17%) (Riloff & Shepherd 1997) and 93/257 (36%) (Roark & Charniak 1998). So our results are significantly better than those reported previously for this category. To our knowledge, no one has reported results for the other categories that we tested.

We also wanted to verify that the phrases in the semantic lexicon would be likely to appear in new texts. So we created a test set by manually tagging all noun phrases that were legitimate extractions for each category in 233 new web pages.[5] Table 2 shows the recall and precision scores on the test set for three experiments. In the first experiment (*Baseline*), we generated a baseline by extracting all noun phrases in the test set that contained one of the original seed words. In the second experiment (*Lexicon*), we manually filtered the semantic lexicon to remove incorrect entries and then extracted every noun phrase in the test set that contained a lexicon entry. In the third experiment (*Union*), we extracted all noun phrases in the test set that either contained a lexicon entry or were extracted by an extraction pattern generated for the category.

Recall/Precision (%)	*Baseline*	*Lexicon*	*Union*
Web Company	10/32	18/47	18/45
Web Location	11/98	51/77	54/74
Web Title	6/100	46/66	47/62

Table 2: Recall/Precision (%) Results on Web Test Set

Table 2 shows that the seed words by themselves achieved high precision for locations and titles but low recall. The low precision for companies is mainly due to the presence of "company" in the seed word list, which is extremely common but extracts mostly referents and not specific company names. We did not count referents as legitimate extractions in these experiments.

The second column (*Lexicon*) shows that the semantic lexicons were useful for extracting information from new web pages. The lexicon achieved about 50% recall with 66-77% precision for locations and titles. The results for companies were substantially lower, but still above the baseline. We hypothesize that the set of possible company names is much larger than the set of

[5]Due to time constraints, we only hand-labeled the noun phrases that were extracted by at least one of the 19,690 candidate patterns produced by AutoSlog.

locations and titles in corporate web pages, so we probably need to generate a much larger lexicon of company names to achieve good results for this category.

The third column (*Union*) shows that using the lexicon and the extraction patterns to identify new information slightly increases recall for locations and titles, but also slightly decreased precision. In retrospect, we realized that we probably need to use more extraction patterns. For this experiment, we only used patterns with a score > 0.7, which produced only 63 title extraction patterns and 87 company extraction patterns. Since the patterns represent very specific linguistic expressions, we probably need to lower that threshold. We also plan to consider schemes for allowing both the semantic lexicon and the extraction patterns to vote on possible extractions.

Conclusions

Bootstrapping is a powerful technique for leveraging small amounts of knowledge to acquire more domain knowledge automatically. An important aspect of our bootstrapping mechanism is that it generates domain-specific dictionaries. For example, the location dictionary generated from the web pages contained mainly country names and U.S. cities while the location dictionary generated from the terrorism articles contained mostly cities and towns in Latin America. Generating domain-specific dictionaries is a strength because the dictionaries are tailored for the domain of interest. But some categories may behave strangely if one does not anticipate their role in the domain. For example, we tried using this bootstrapping technique for the semantic category "vehicle" using the terrorism corpus, but the resulting dictionaries looked remarkably similar to the weapon dictionaries. In retrospect, we realized that vehicles are often weapons in the terrorism texts, either as car bombs or fighter planes. So in this domain, considering vehicles to be weapons usually makes sense.

In summary, we have shown that multi-level bootstrapping can produce high-quality dictionaries for a variety of categories. Our bootstrapping method has two advantages over previous techniques for learning information extraction dictionaries: both a semantic lexicon and a dictionary of extraction patterns are acquired simultaneously, and no special training resources are needed. Our algorithm needs only a corpus of (unannotated) training texts and a small set of seed words as input. The resulting semantic lexicon does need to be manually inspected to get rid of bad entries, but this can be done in a few minutes. Multi-level bootstrapping appears to be a promising approach for acquiring domain knowledge automatically, and we hope to apply this technique to other knowledge acquisition tasks as well.

Acknowledgments

This research is supported in part by the National Science Foundation under grants IRI-9509820, IRI-9704240, and SBR-9720374, and the DARPA HPKB program under contract F30602-97-1-0215.

References

Califf, M. E. 1998. *Relational Learning Techniques for Natural Language Information Extraction.* Ph.D. Dissertation, Tech. Rept. AI98-276, Artificial Intelligence Laboratory, The University of Texas at Austin.

Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A.; Mitchell, T.; Nigam, K.; and Slattery, S. 1998. Learning to Extract Symbolic Knowledge from the World Wide Web. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence.*

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

Huffman, S. 1996. Learning information extraction patterns from examples. In Wermter, S.; Riloff, E.; and Scheler, G., eds., *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing.* Springer-Verlag, Berlin. 246–260.

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

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

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

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

Riloff, E. 1993. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 811–816. AAAI Press/The MIT Press.

Riloff, E. 1996a. An Empirical Study of Automated Dictionary Construction for Information Extraction in Three Domains. *Artificial Intelligence* 85:101–134.

Riloff, E. 1996b. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 1044–1049. The AAAI Press/MIT Press.

Roark, B., and Charniak, E. 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*, 1110–1116.

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

Soderland, S. 1999. Learning Information Extraction Rules for Semi-structured and Free Text. *Machine Learning.* (to appear).