A Retrospective on Mutual Bootstrapping

Ellen Riloff, Rosie Jones

■ *This retrospective article discusses the mutual bootstrapping technique for weakly supervised learning of extraction patterns and semantic lexicons from unstructured text, originally published in AAAI'99 (Riloff and Jones 1999). We present an overview of the mutual bootstrapping approach, describe related research that has followed from the original work, and discuss lessons learned about bootstrapped learning for natural language processing.*

When we were invited to write a retrospective article about our 1999 AAAI conference paper on mutual bootstrapping (Riloff and Jones 1999), our first reaction was hesitation because, well, that algorithm seems old and clunky now. But upon reflection, we realized that this early work shaped a great deal of subsequent work on bootstrapped learning for natural language processing, both by ourselves and others. So our second reaction was enthusiasm for the opportunity to think about the path from 1999 to 2018 and to share the lessons we've learned about bootstrapped learning along the way.

This article begins with a brief history of related research that preceded and inspired the mutual bootstrapping work, to position it with respect to that period of time. We then describe the general ideas and approach behind the mutual bootstrapping algorithm. Next, we survey several types of research that have followed and that share similar themes: multiview learning, bootstrapped lexicon induction, and bootstrapped pattern learning. Finally, we discuss some of the general lessons that we have learned about bootstrapping techniques for natural language processing (NLP) to offer guidance to researchers and practitioners who may be interested in exploring these types of techniques in their own work.

Background

The early 1990s marked a sea change in the natural language processing community as statistical methods and machine learning techniques took hold of imaginations and rapidly proliferated. Researchers began collecting large text corpora, annotating data, computing probabilities and statistical metrics, and training machine learning classifiers. Research in information extraction (IE) was a prime example of the statistical revolution, as people enthusiastically transitioned from systems built upon handcrafted rules and patterns to systems that learned rules and patterns from manually annotated data or handcrafted knowledge (for example, AutoSlog [Riloff 1993], PALKA [Kim and Moldovan 1993], CRYSTAL [Soderland et al. 1995], LIEP [Huffman 1996], RAPIER [Califf 1998], SRV [Freitag 1998], WHISK [Soderland 1999]).

The term e*xtraction pattern* was coined to describe a lexico-syntactic context surrounding a noun phrase (NP) that describes its relationship to an event, an entity, or a concept. These patterns were originally used for information extraction tasks related to events. For example, some of the patterns generated by AutoSlog for WEAPONS were *<Subject> was hurled, confiscated <DirectObject>*, and *explosion of <NP>*, where the WEAPON term (for example, "grenade") occurred in the bracketed position. This notation is shorthand for the sake of readability, but the actual patterns have to match a specific syntactic structure based on a shallow parse of the sentence. For example, the pattern *<Subject> was hurled* requires the word *hurled* to be in a passive voice verb phrase, and the subject of *hurled* is the extracted NP.

While AutoSlog learned from a manually annotated text collection, AutoSlog-TS (Riloff 1996) succeeded AutoSlog as a weakly supervised extraction pattern learner that required only (unannotated) relevant and irrelevant documents as input. AutoSlog-TS exhaustively generates a pattern to extract every NP in the corpus, and then ranks the patterns based on the strength of their association with the relevant documents. In a final step, a human reviews the patterns to assign types and ensure their integrity.

Although all of these methods required human supervision in the form of annotated data, knowledge, or manual review, the learned IE systems took far less effort to build than the previous generation of IE systems. Before these learning methods, IE system developers would spend months, if not years, handcrafting rules and patterns, assessing their performance on individual documents through manual observation, and continually refining them to improve system performance. This process was painstaking and time intensive, so the prospect of automating the task was a welcome advance. The proverbial knowledge-engineering bottleneck that had plagued NLP systems was beginning to show cracks.

In 1995, David Yarowsky published a seminal paper about bootstrapped learning for word sense disambiguation. This paper showed how a few seed words could jumpstart a learning process that iteratively improved classification performance without additional human supervision. This idea was utterly captivating, offering the possibility of learning semantic knowledge with just a tiny amount of human input. Bootstrapped learning with only initial human seeding had the potential to finally shatter the knowledge-engineering bottleneck.

Inspired by Yarowsky's work, Ellen Riloff's research group began exploring bootstrapped learning to create semantic dictionaries. For many NLP tasks, it is important to know the semantic types of words, for example that a *dog* is an ANIMAL, a *sofa* is FURNITURE, and a *truck* is a VEHICLE. Of course, words have richer semantics than this, but NLP systems often lack even this basic semantic knowledge. Around the same time, WordNet (Miller 1990) was created as a lexical resource of English words and their senses organized in a semantic hierarchy. WordNet quickly became a highly used resource, and still is today. But information extraction systems typically focus on a specific domain, and general-purpose resources are rarely sufficient for text processing in specialized domains because of domain-specific terminology and biases. The domain of veterinary medicine, to take one example, is characterized by medical vocabulary (for example, *diseases, symptoms, drugs*), idiosyncratic abbreviations (for example, *abx* for antibiotics, *pred* for prednisone, *dachsie* for dachsund), and atypical word senses (for example, *mix* and *cross* commonly refer to animals, as in *terrier/beagle cross*) (Huang and Riloff 2010).

When an NLP system is being designed to process texts in a specific topic area, then it is extremely beneficial to have a semantic dictionary tailored for that domain. One reason is that domain-specific vocabulary is often missing from general resources. Another reason is that the most common sense for a word in general texts may be different from the dominant sense of the word in a specific domain. For example, a *bat* is nearly always an instrument in baseball articles, but often refers to a flying mammal in other types of articles. Even familiar domains that frequently appear in the news often have domain-specific vocabulary that cannot be adequately covered by general-purpose resources. For example, sports articles frequently mention sports teams, which often have unique names such as *Phillies, Lakers,* or *Knicks,* or they are named after other types of entities such as animals (for example, *Tigers*, *Eagles, Panthers*). As another example, news articles about terrorism frequently mention terrorist organizations (for example, *ETA, FARC, FMLN*) and numerous types of weapons (for example, *arms, AK-47, M-16*).

The task of automatically creating domain-specific semantic dictionaries is called *semantic lexicon induc-*

tion. The early algorithms for bootstrapped semantic lexicon induction exploited the observation that semantically related nouns often co-occur in conjunctions (for example, *lions and tigers and bears*), lists (for example, *lions, tigers, bears ...*), appositives (for example, *stallions, male horses*), and compound nouns (for example, *tuna fish*). These bootstrapping algorithms began with a few seed nouns for the semantic category of interest and identified nouns that co-occurred with the seeds, in close proximity or in specific syntactic constructions (Riloff and Shepherd 1997; Roark and Charniak 1998). The resulting dictionaries were far from perfect, but they showed that bootstrapped learning of semantic dictionaries was possible and that this approach was a promising direction to pursue.

The mutual bootstrapping algorithm described in the next section brought together these seemingly different tasks of learning information extraction patterns and learning semantic dictionaries. Mutual bootstrapping showed that both types of knowledge could be learned at the same time, and that doing so was mutually beneficial.

Mutual Bootstrapping

The mutual bootstrapping algorithm in our 1999 AAAI conference paper hinged on two key ideas: (1) words and contextual patterns can be used independently to identify instances of a semantic category, and (2) multiple knowledge sources can serve as the foundation for bootstrapped learning when played off each other. The process of simultaneously learning two different types of knowledge by alternately leveraging one type of knowledge to learn the other was called *mutual bootstrapping*.

In our work, we identified two types of knowledge that are often sufficient to identify the semantic category of a noun phrase (NP) in context: (1) the head noun of the NP itself, and (2) the context surrounding the NP. For example, consider the sentence: *The brown dog barked at the cat*. We can infer that *The brown dog* is an ANIMAL in two ways: (1) because we know that the word *dog* commonly refers to ANIMALS, or (2) because *barking* is an action usually performed by ANIMALS. Of course, there are exceptions, for example a *hot dog* often refers to FOOD, and people can also bark when they are angry. But either type of knowledge, lexical or contextual, is usually sufficient by itself to make a strong inference about the meaning of a phrase in context.

Figure 1 illustrates the mutual bootstrapping learning process. The bootstrapping cycle begins with a text corpus and a small set of manually defined seed words for the targeted semantic category. For example, suppose you want to learn words and patterns for the semantic category DISEASE. The input would consist of a large collection of texts that frequently mention diseases and a small set of seed words that refer to diseases, such as *cholera, flu, listeria, measles,* and *tuberculosis*. The seed words are used as the initial semantic lexicon, which is iteratively expanded during the bootstrapping process.

The AutoSlog pattern generator (Riloff 1993) was then applied to the text corpus in an exhaustive fashion to produce an extraction pattern for literally every noun phrase (NP) in the corpus. This process produces an enormous set of patterns paired with the NPs that they extract, which collectively represent all of the noun phrase contexts in the corpus. The mutual bootstrapping algorithm uses this data, along with the initial semantic lexicon, both to induce a pattern dictionary for the semantic category and to grow the semantic lexicon. The learning process has two alternating steps. First, all of the patterns are scored based on their strength of association with the terms in the semantic lexicon (for details, see Riloff and Jones 1999). The highest-scoring pattern is added to the pattern dictionary. Second, in a leap of faith, all of the NPs extracted by the newly added pattern are assumed to belong to the targeted semantic category and their head nouns are added to the semantic lexicon. The process then iterates: all of the patterns are rescored based on the expanded semantic lexicon, a new pattern is selected and added to the pattern dictionary, its extracted head nouns are added to the lexicon, and so on.

To return to figure 1, imagine that the pattern *infected with <NP>* is the highest-scoring pattern because many of the seed terms occur in that context. This pattern would be added to the pattern dictionary, and the head nouns of all NPs that occurred in this pattern context are assumed to be DISEASES and added to the semantic lexicon. In the example, the newly added terms would be *ebola, malaria, plague, pneumonia,* and *tularemia*. The expanded semantic lexicon then serves as a larger set of seed terms for the next iteration of bootstrapping.

While this approach worked well in many cases, one problem was the leap of faith that all NPs extracted by a pattern belong to the same semantic category. Even when strongly associated with a category, contextual patterns can co-occur with NPs of different semantic categories because very few contexts occur with a single semantic category 100 percent of the time. Occasional incorrect lexicon entries typically will not change the course of bootstrapping very much, especially if the errors are distributed across different competing categories (that is, if the errors represent a variety of different semantic classes). But a serious problem exists when many incorrect lexicon entries belong to the same (incorrect) semantic class. This phenomenon arises when some contexts systematically occur with multiple semantic categories. For example, locations and temporal expressions both frequently occur with event phrases and the preposition *in* or *on,* such as *happened in (Boston/November)* and *occurred on (Wall Street/Mon-*

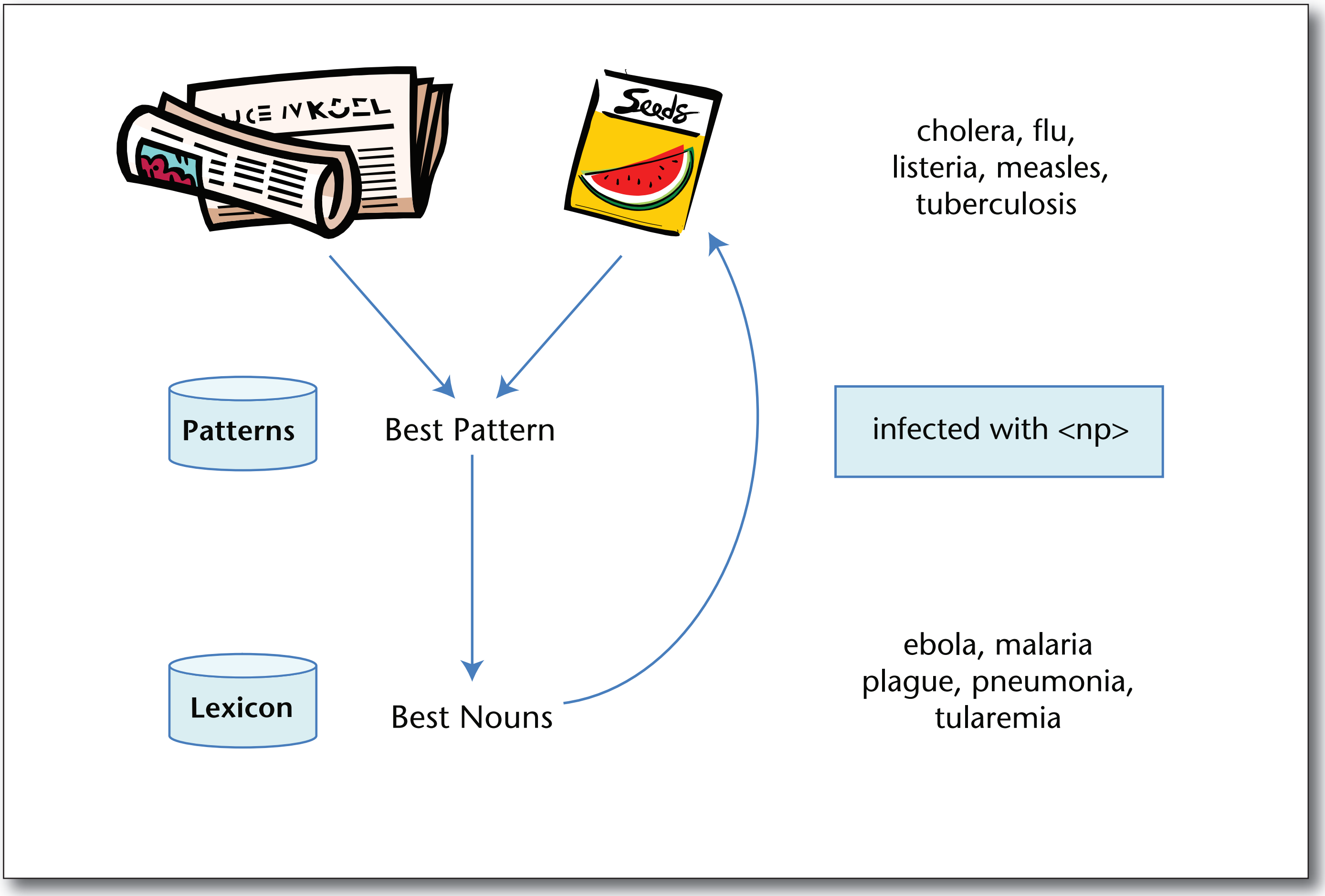

Figure 1: The Mutual Bootstrapping Process.

day). As another example, weapons, people, and natural disasters frequently occur in expressions that describe the cause of injuries or damage, such as *a (grenade/sniper) killed three people* or *a (bomb/tornado) caused massive damage*.

To address this problem, we added a second layer of bootstrapping, called *meta-bootstrapping*. After the mutual bootstrapping process completed, the learned lexicon entries were reevaluated and only the five most trusted entries were retained. The mutual bootstrapping process was then restarted using the five new category members as additional seed words. The criteria for reevaluating the lexicon entries was based on the number of different patterns that occurred with the term.

The meta-bootstrapping solution was a bit ad hoc and it was expensive because the learning process now involved nested bootstrapping processes. Since then, better solutions have been found that induce semantic dictionaries based on multiple contextual patterns, rather than a single pattern, and that can detect semantic drift during bootstrapping. Nevertheless, the mutual bootstrapping idea at the heart of this work has proven to be useful in a variety of subsequent research efforts, including research on semantic lexicon induction and other tasks, as we discuss in the next section.

Subsequent Work

In the years since our mutual bootstrapping paper, a wide variety of related research has appeared. Here, we present a brief overview of some of the most closely related work that has emerged, with the caveat that this summary aims to highlight different avenues of follow-on work and as such, it is not intended to be a comprehensive literature survey.

Learning with Multiple Views

An important aspect of the mutual bootstrapping algorithm is that it uses two facets of the data, the noun phrases and their contexts, to learn from a small initial set of seeds. The idea of learning from multiple knowledge sources also arose contemporaneously in Collins and Singer's (1999) work on bootstrapped learning for named entity recognition and in Blum and Mitchell's (1998) work on cotraining. Collins and Singer's procedure for named entity recognition (NER) is an iterative learning process that exploits the same types of dual knowledge sources as the mutual bootstrapping algorithm did. Specifically, their NER system learned by exploiting the fact that named entities can often be classified into types based on (1) the presence of specific words in the phrase (for example, *Contains*(*X*, *"Mr."*) ⇒ PERSON(X)) or (2) the context surrounding the phrase (for example, "CEO of <X>" ⇒ ORGANIZATION(X)). However, their work focused on learning a classifier to assign types to named entities in context, while the mutual bootstrapping work produced two dictionaries, a semantic lexicon and a set of patterns.

For the document classification task, cotraining (Blum and Mitchell 1998) used two different classifiers to train each another, starting from a few seed training instances, based on the observation that classifiers capturing different views of the data tend to be complementary and make independent errors. Under independence assumptions, strong performance guarantees can be provided for this algorithm. Cotraining was later used for semantic lexicon induction by capturing multiple views based on different types of syntactic constructions (Phillips and Riloff 2002). More recently, Mitchell et al. (2015) extended the cotraining insight to include large numbers of different learning tasks, whose complementary natures help improve precision and prevent semantic drift. Other lines of work have examined the mutual learning of relations and the concepts that fill those relations (for example, Bing et al. [2017]).

It is tempting to use self-training algorithms, which train a classifier from a small set of initial data, apply the classifier to unlabeled data, and then retrain themselves based on their predictions in an iterative loop. However, Nigam and Ghani (2000) showed that self-training, for example with expectation maximization (EM), does not always work without the two complementary data views. In our experience, a typical self-training process will often improve the recall of a classifier, but with a comparable, or sometimes larger, drop in precision.

Bootstrapping Semantic Lexicons

The original mutual bootstrapping algorithm hypothesized new category members based on a single contextual pattern. This strategy was identified as a source of errors that led to the addition of a second layer of meta-bootstrapping to selectively choose the best candidates from among the induced terms and then restart the entire bootstrapping process.

For semantic lexicon induction, the immediate successor to the mutual bootstrapping algorithm was another bootstrapped learner called Basilisk (Thelen and Riloff 2002). Basilisk introduced two new ideas that improved the quality of lexicon induction: (1) learning based on collective evidence over a set of contextual patterns, and (2) learning multiple semantic classes simultaneously. Rather then rely on a single contextual pattern to generate category members during bootstrapping, Basilisk identifies a set of contextual patterns that frequently co-occur with known category members. All terms in these contexts are considered as candidates for the lexicon. The candidates are then scored based on all of the contexts in which they occur, and the best candidates are added to the lexicon. With the focus on an aggregate contextual profile, a term is selected as a category member only if it consistently occurs in the same types of contexts as known category members.[1] Basilisk's bootstrapping process is depicted in figure 2.

The second novelty of Basilisk was the idea that learning multiple semantic classes simultaneously can help to constrain and steer the bootstrapping process. Since the learning process is fully automatic with no human feedback, bootstrapped learners can easily stray from the original target concept. But if you are willing to make the assumption that a term can belong to only one semantic class, then learning multiple semantic classes at the same time can provide much-needed guidance because terms learned for one class serve as negative examples for the other classes. In general, words are highly polysemous in natural language, so this assumption is usually not valid. But the main motivation for bootstrapped semantic lexicon induction was to enable the rapid creation of semantic dictionaries for specialized domains. Within the confines of a limited domain, most words have a dominant meaning, so in this context a one-sense-per-domain assumption is quite reasonable. Basilisk uses this assumption to score each pattern based on its co-occurrence with both category members (positive examples) and noncategory members (negative examples). The negative examples help the learner recognize when it is drifting into another category's territory.

McIntosh and Curran (2008) later developed a similar bootstrapped learner for semantic lexicon induction called WMEB (Weighted Mutual Exclusion Bootstrapping). Their goal was to generate semantic lexicons for the biomedical domain for categories such as cells, mutations, and tumors. Their work used N-gram pattern contexts (specifically, 5-grams) to avoid the need for syntactic analysis. Researchers have observed that bootstrapped learners can be prone to semantic drift (Komachi et al. 2008; McIntosh and Curran 2009; Vyas and Pantel 2009), where the terms learned for one category gradually drift toward a different category. This drift can occur when there is systematic polysemy of terms across two categories. For example, celestial bodies are often named after Roman Gods (Vyas and Pantel 2009). As mentioned earlier, drift can also happen when certain types of contexts systematically occur with multiple semantic categories. McIntosh and Curran (2009) devised a clever solution to proactively identify semantic drift by comparing the distributional similarity of new candidates both with recently learned

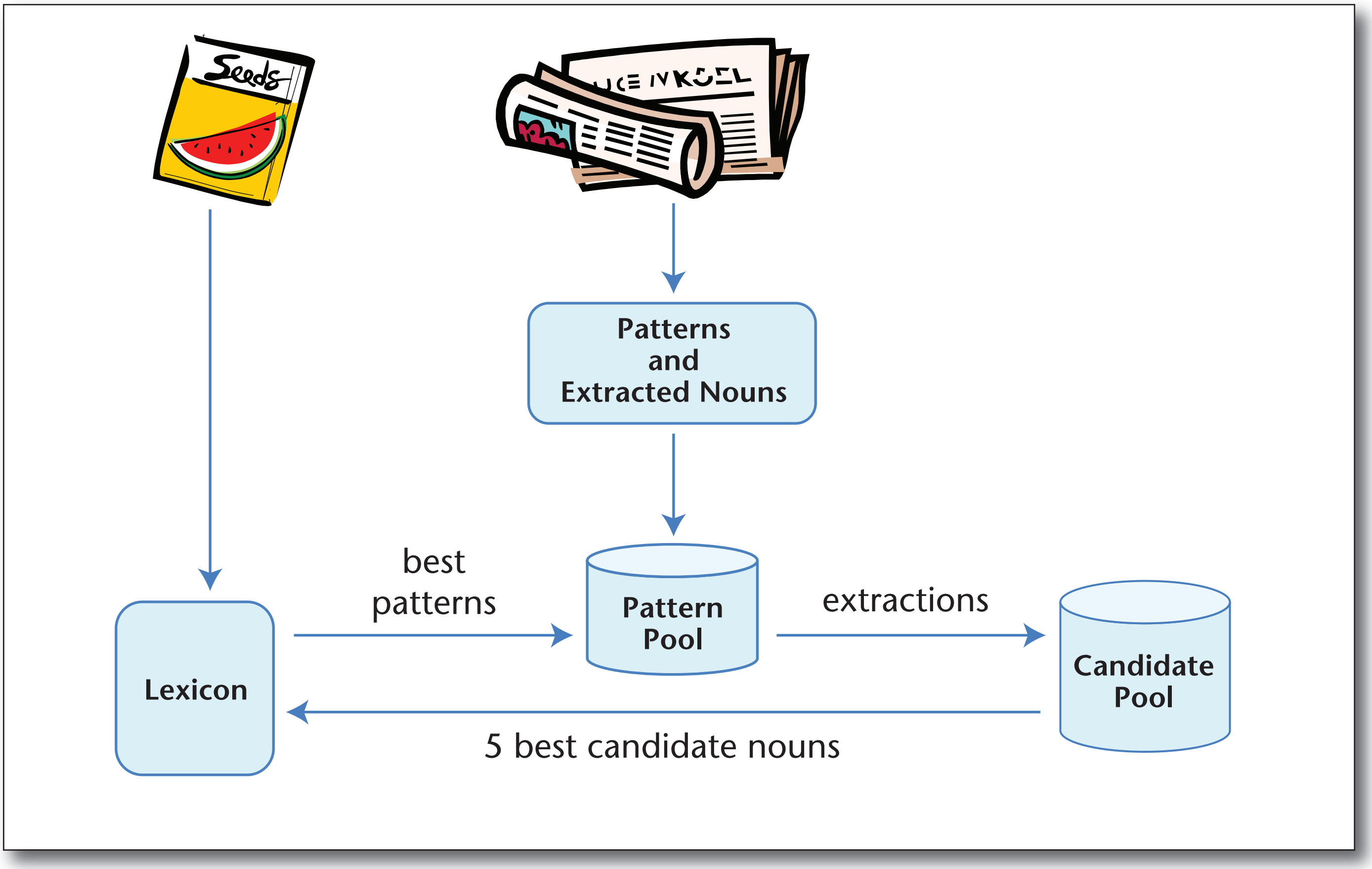

Figure 2: The Basilisk Algorithm.

terms and with terms learned in the early iterations of bootstrapping. Candidates that are more similar to recently learned terms than to earlier terms are likely to have drifted.

Subsequent research has also recognized the value of learning multiple categories during bootstrapping as a source of negative feedback. McIntosh and Curran (2010) developed a method to automatically discover new semantic categories that can be beneficial as negative examples during learning. Vyas and Pantel (2009) also recognized the importance of identifying negative classes in their work on set expansion. They had a human manually identify errors and automatically removed additional words that had high distributional similarity with the errors, under the hypothesis that they were likely to belong to the same negative class. Learning multiple categories simultaneously has also been shown to be useful as counter-training for pattern learning (Yangarber 2003) and for cross-category learning in a bootstrapped contextual semantic tagger (Huang and Riloff 2010).

Bootstrapping New Types of Lexicons

Natural language processing systems need many types of knowledge, and the same bootstrapped learning mechanisms used to create semantic dictionaries have proven to be beneficial for creating other types of dictionaries as well.

In the years since the Basilisk algorithm was developed for semantic lexicon induction, Basilisk has been used to generate several novel types of lexicons. Given a seed list of subjective nouns, which represent private states and opinionated language, Basilisk learned to identify many new subjective nouns to produce a subjectivity lexicon (Riloff, Wiebe, and Wilson 2003; Wilson et al. 2005). Examples of the subjective nouns learned are *barbarian, atrocities,* and *exaggeration*. For event extraction, Basilisk learned to identify role-identifying nouns, which are nouns whose semantics reveal the role of an entity with respect to an event (Phillips and Riloff 2007). For example, words such as *assassin, burglar*, and *sniper* refer to people who participated as the agent of an event, while *casualty, fatality,* and *victim* refer to people who represent the patient of an event.

For work on generating plot unit representations (Goyal, Riloff, and Daumé III 2010, 2013), Basilisk was used to identify patient polarity verbs (PPVs), which are verbs that affect their patients in positive or negative ways. For example, being *fed, paid,* or *adopted* are typically desirable events for the entities that are acted upon, but being *eaten, chased,* or *hospitalized* are usually undesirable events. Basilisk learned to identify many PPVs using separate bootstrapping processes to learn positive and negative PPVs, given 10 manually defined examples of each. The contextual patterns, however, were quite different from the lexico-syntactic patterns used in previous Basilisk work. For the task of learning PPVs, conjunction patterns were defined to exploit a previous observation from sentiment analysis that conjunctions usually join items with the same polarity (Hatzivassiloglou and McKeown 1997). For example, if *rescued* is known to be a positive verb, then seeing the conjunction *rescued and adopted* should lead us to believe that *adopted* is probably also a positive verb.

Research on sarcasm recognition by Riloff et al. (2013) demonstrated that different types of lexicons can be learned simultaneously if they co-occur in a shared structure. This work used just one seed word, *love,* as an example of a positive sentiment, and a collection of tweets that contain a #sarcasm hashtag. The key idea behind this work is that sarcasm often emerges from the juxtaposition of a positive sentiment and a negative situation (for example, *I love being ignored* or *I just adore waiting for the doctor)*. Given a sarcastic tweet that has a positive sentiment (for example, *love)*, we can infer that the target of the sentiment is probably a negative situation. Conversely, given a sarcastic tweet that mentions a negative situation (for example, *being ignored)*, we can infer that the sentiment is probably positive. Using these dual sources of knowledge in an alternating bootstrapping cycle, the system learned lists of positive sentiments and negative situations simultaneously.

These research efforts illustrate the flexibility and generality of bootstrapped lexicon induction. Given different types of seed words, patterns, and corpora, this paradigm can be used to learn many different kinds of knowledge.

Bootstrapped Pattern Learning

Bootstrapping has also been used to learn patterns in several novel ways. The Ex-Disco system (Yangarber et al. 2000) added a bootstrapping mechanism around a pattern learner modeled after AutoSlog-TS, which used relevant and irrelevant texts for training. Ex-Disco used a small set of manually defined seed patterns to heuristically partition a collection of unannotated texts into relevant and irrelevant sets. Patterns were then ranked based on their association with the relevant texts, the best pattern(s) were added to the pattern set, and the corpus was repartitioned into new relevant and irrelevant sets for the next iteration. Stevenson and Greenwood (2005) also began with seed patterns and used semantic similarity measures to iteratively rank and select new candidate patterns based on their similarity to the seeds.

Riloff and Wiebe (2003) used a bootstrapping mechanism to learn lexico-syntactic patterns that represent subjective expressions. A novel aspect of that work was the use of high-precision (but low-recall) classifiers as the basis for seeding. Lukin and Walker (2013) adopted a similar bootstrapping approach to learn patterns associated with sarcasm in dialogue. Recently, Gupta and Manning (2014) used several types of unsupervised class predictors, such as distributional similarity, to better estimate the likelihood that an unlabeled term belongs to a negative class during pattern scoring. They also showed that incorporating distributed word representations to enhance the training sets during learning can improve results (Gupta and Manning 2015).

Lessons Learned

Bootstrapped learning can seem like a black art to people who do not yet have experience with it. But many lessons have been learned by researchers who work in this paradigm. In this section, we discuss some of the most important (but often unspoken) principles behind bootstrapped learning techniques, extrapolating both from the literature and from our own personal experiences.

Seeding Principles

In NLP, the paradigm of bootstrapped learning takes human input in the form of seeding heuristics and applies those heuristics to unannotated texts to produce labeled examples. The heuristically labeled data is then used to train a learning algorithm, which kicks off an iterative process.

Since the seeding heuristics are a proxy for manually labeled data, it is critical that they be able to assign labels with reasonably high accuracy. Heuristics will rarely be as accurate as human annotators, but they can be applied automatically to large volumes of unannotated text, yielding a potentially enormous amount of labeled training data. The expectation is that large volumes of slightly noisy training data will be nearly as good as, or potentially better than, presumably smaller amounts of "perfectly" labeled training data (which is typically available only in limited quantities). With this in mind, we lay out three general principles for identifying effective seeding heuristics: high frequency, high precision, and diversity.

High Frequency

Simply put, you want the seeding heuristics to be able to label as many examples as possible. The more instances they label, the more training data the learning algorithm gets. Many people have observed that using different seeds can produce dramatically different results (for example, McIntosh and Curran 2009). One of the reasons is that different seeds can result in vastly different amounts of labeled data. Consequently, a learner trained with one set of seeds may have, say, twice as much training data as a learner trained with a different set of seeds. A common mistake is for people to identify seeds by hand, assuming that they know what the most frequent words will be. Corpora are highly idiosyncratic, and words that may often be common will not necessarily be common in any particular text collection. Furthermore, people can miss words that are extremely common in the corpus and that could therefore be valuable seeds. Looking at corpus statistics is the most reliable way to know that your seeds are aligned with your actual data.

For lexicon induction, we typically follow a procedure first suggested by Roark and Charniak (1998) to ensure high-frequency seeding. To select seed words, sort all of the candidates (for example, nouns) in the unannotated text corpus by frequency. Then manually review the list, selecting the first (most frequent) *k* words that belong to the category of interest. This process ensures that the seeds will be highly frequent in the corpus.

High Precision

The seeding heuristics are used to automatically label data for training, so the accuracy of the heuristics directly correlates with the quality of the training data. Consequently, the heuristics should have high precision. However, high precision can be at odds with high frequency because there is often a trade-off between precision and recall (that is, high-precision rules are often low recall, and vice versa). Consequently, it is sometimes difficult to identify seeding heuristics that satisfy both the high-precision and high-frequency goals. When forced to choose, high precision is usually preferable for two reasons: (1) too much noise in the training data can render it ineffective, and (2) low recall can often be compensated for by increasing the size of the text corpus. Since only unannotated texts are needed, obtaining more texts is often feasible.

Diversity

The seeding heuristics should be able to label a diverse set of examples, so as to produce labeled data that is (reasonably) representative of the corpus as a whole. If the seeding heuristics only label instances that are highly similar, or if they have poor coverage across subclasses, then the resulting data will be strongly biased. Of course, by definition the labeled instances will share whatever properties are selected for by the seeding mechanism. But diversity can often be achieved by defining seeding heuristics that are not too specific and by using multiple seeding heuristics that cover different parts of the search space. It has been shown that if we characterize the corpus in terms of connectivity of seeds and contexts as a graph, then we should try to cover any subgraphs disconnected from the main graph (Jones 2004).

Seeding Mechanisms

Bootstrapped learning has been applied to a wide variety of NLP tasks, using many types of seeding strategies. Here we take a brief look at some of the seeding mechanisms that have been used successfully, to emphasize that bootstrapping can be initiated in many different ways.

Seed words are a common form of seeding that has been used for lexicon induction, pattern learning, and word sense disambiguation (Yarowsky 1995). Seed patterns have been used to identify relevant contexts, for example to classify relevant and irrelevant texts for bootstrapped pattern learning (Yangarber et al. 2000) and to identify relevant regions to train a sentence classifier (Patwardhan and Riloff 2007). *Seeding rules* have been used for named entity recognition and coreference resolution. Collins and Singer (1999) heuristically labeled named entities to create training data by defining rules such *Contains*(*X*, *"Mr."*) ⇒ PERSON(X). Bean and Riloff (2004) used lexical and syntactic seeding rules to generate labeled data for coreference resolution.

Another approach is to create an initial seeding classifier that is applied to unlabeled texts to produce an initial set of labeled instances, which are then used as training data to jumpstart a bootstrapped learning process. This scenario makes sense when it is possible to easily construct a high-precision, but potentially low-recall, classifier. This type of classifier can initially label some instances with high accuracy, which the bootstrapping process can use to learn new information. This approach has been used for opinion analysis, to learn patterns representing subjective expressions (Riloff and Wiebe 2003) and to train subjective and objective sentence classifiers without annotated data (Wiebe and Riloff 2005).

It is worth noting that *distant supervision* is also a seeding method, although it is typically used to generate a large set of labeled data to train a supervised learner in a single step. Distant supervision takes advantage of an existing knowledge base (KB) to heuristically label instances that correspond to data found in the KB. For example, distant supervision has been applied to relation extraction by identifying pairs of entities listed in a knowledge base as having a relation, and then heuristically labeling instances of the entity pairs that appear in close proximity as positive instances of the relation (Mintz et al. 2009).

Secondary Benefits of Bootstrapping

The primary benefit of bootstrapped learning is that it eliminates the need for manually annotated training data, which is expensive and time consuming to obtain. However, bootstrapping methods have several secondary benefits as well, which are often underappreciated.

First, bootstrapped learning allows for easier and more freewheeling system design, development, and experimentation. Since supervised learning depends on manually annotated data, system development often must wait until annotated data has been collected. And then, system developers are handcuffed to that data set because it is the only available training data. In contrast, with bootstrapped learning systems, the only input consists of unannotated texts for the domain and a seeding mechanism. Unannotated text corpora are relatively easy to obtain, and most seeding strategies are lightweight. As a result, it is blissfully easy to try bootstrapped learning on different domains, text corpora, and tasks.

For example, suppose someone is interested in semantic lexicon induction for a new domain. If the person has an interest in creating an NLP system for the domain, then they probably already have (or know where to find) a large collection of texts for that domain. Given the text corpus, the person needs only to define a small number of seed terms for each semantic category. A key question is what the ideal set of semantic categories should be. That's where the benefits of the bootstrapping paradigm become apparent. The developer can choose an initial set of categories based on their domain knowledge and define a small set of seed words for each one. This process may take as little as an hour. Then the developer can apply the learning algorithm and inspect the results. If new words are learned that clearly belong to different categories, then new categories can be added simply by defining a few seed words for them. If some categories are behaving similarly, then the developer may choose to merge categories to represent a higher-level concept. If the frequencies look small, the developer can expand the size of the corpus simply by obtaining more unannotated texts. Furthermore, cross-resource experimentation is also relatively straightforward. Experimenting with different text corpora and even languages (depending on the task) can be as simple as replacing one text collection with another and mapping the seeding strategy onto the new resource. While these changes may not be trivial, explorations like these are substantially easier in a bootstrapping paradigm than they would be in a supervised learning paradigm that requires manually annotated training data.

Another advantage of learning with seeding strategies as opposed to manually annotated data is that the former is typically easier for people to produce than the latter. In natural language processing, manually annotating texts can be deceptively difficult because of issues pertaining to phrase boundaries, edge cases (borderline concepts), and idiosyncratic expressions. Given any set of natural language documents, many of these issues are likely to appear and can be impossible to avoid. Overall, bootstrapped learning offers many advantages, from the perspective of data requirements as well as research and system development efforts.

Summary

Nineteen years after appearing in the AAAI'99 conference, our paper continues to be cited. As we have tried to show, we did not start the revolution alone, but were part of a movement that has continued to have an impact on research today.

It is difficult to believe that the long-term success of natural language processing will rely on manually annotated text corpora for every conceivable task, domain, and language. Bootstrapping, weakly supervised learning, and distant labeling are important tools for the future, especially as text corpora continue to grow in size, massive computing power becomes increasingly available to support large-scale text processing, and NLP applications are ever more ubiquitous in everyday life.

There remain many open questions and research avenues for future work, both within natural language processing in general and for bootstrapped learning methods in particular. Accuracies are still far from perfect for many NLP tasks, and new applications for NLP are constantly emerging. Our hope is that the next generation of researchers will continue investigating and improving bootstrapping learning methods for natural language processing and that these techniques will play a major role in future NLP technologies.

Notes

[1] This is essentially a form of distributional similarity, which has become a widely used NLP tool for empirical semantic analysis.

References

Bean, D., and Riloff, E. 2004. Unsupervised Learning of Contextual Role Knowledge for Coreference Resolution. In *Proceedings of the 42nd Annual Meeting of the North American Chapter of the Association for Computational Linguistics* (HLT/NAACL 2004). Stroudsburg, PA: Association for Computational Linguistics.

Bing, L.; Dhingra, B.; Mazaitis, K.; Park, J. H.; and Cohen, W. W. 2017. Bootstrapping Distantly Supervised IE Using Joint Learning and Small Well-Structured Corpora. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence,* 3408–3414. Palo Alto, CA: AAAI Press.

Blum, A., and Mitchell, T. 1998. Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory* (COLT'98), 92–100. New York: Association for Computing Machinery. doi.org/10.1145/279943.279962

Califf, M. E. 1998. Relational Learning Techniques for Natural Language Information Extraction. PhD dissertation. Technical Report AI98-276, Artificial Intelligence Laboratory, The University of Texas at Austin.

Collins, M., and Singer, Y. 1999. Unsupervised Models for Named Entity Classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora* (EMNLP/VLC'99). Stroudsburg, PA: Association for Computational Linguistics.

Freitag, D. 1998. Toward General-Purpose Learning for Information Extraction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/980845.980914

Goyal, A.; Riloff, E.; and Daumé III, H. 2010. Automatically Producing Plot Unit Representations for Narrative Text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2010). Stroudsburg, PA: Association for Computational Linguistics.

Goyal, A.; Riloff, E.; and Daumé III, H. 2013. A Computational Model for Plot Units. *Computational Intelligence* 29(3):466–488. doi.org/10.1111/j.1467-8640.2012.00455.x

Gupta, S., and Manning, C. D. 2014. Improved Pattern Learning for Bootstrapped Entity Extraction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning (CoNLL)*. Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/v1/W14-1611

Gupta, S., and Manning, C. D. 2015. Distributed Representations of Words to Guide Bootstrapped Entity Classifiers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics* (NAACL). Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/v1/N15-1128

Hatzivassiloglou, V., and McKeown, K. 1997. Predicting the Semantic Orientation of Adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, 174–181. Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/976909.979640

Huang, R., and Riloff, E. 2010. Inducing Domain-specific Semantic Class Taggers from (Almost) Nothing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (ACL'10). Stroudsburg, PA: Association for Computational Linguistics.

Huffman, S. 1996. Learning Information Extraction Patterns from Examples. In *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing,* eds. S. Wermter, E. Riloff, and G. Scheler, 246–260. Berling: Springer. doi.org/10.1007/3-540-60925-3_51

Jones, R. 2004. Semi-Supervised Learning on Small Worlds. Paper presented at the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Workshop on Link Analysis and Group Detection (Link KDD 2004), Seattle, WA, August 22–25.

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. doi.org/10.1109/CAIA.1993.366645

Komachi, M.; Kudo, T.; Shimbo, M.; and Matsumoto, Y. 2008. Graph-based Analysis of Semantic Drift in Espresso-like Bootstrapping Algorithms. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2008). Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/1613715.1613847

Lukin, S., and Walker, M. 2013. Really? Well. Apparently Bootstrapping Improves the Performance of Sarcasm and Nastiness Classifiers for Online Dialogue. Paper presented at the Workshop on Language Analysis in Social Media at NAACL HLT, Atlanta, GA, June 11.

McIntosh, T. 2010. Unsupervised Discovery of Negative Categories in Lexicon Bootstrapping. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2010). Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/1687878.1687935

McIntosh, T., and Curran, J. 2008. Weighted Mutual Exclusion Bootstrapping for Domain Independent Lexicon and Template Acquisition. Paper presented at the Australasian Language Technology Association Workshop, Hobart, Australia, December 8–10.

McIntosh, T., and Curran, J. 2009. Reducing Semantic Drift with Bagging and Distributional Similarity. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA: Association for Computational Linguistics.

Miller, G. 1990. Wordnet: An On-Line Lexical Database. *International Journal of Lexicography* 3(4). doi.org/10.1093/ijl/3.4.235

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant Supervision for Relation Extraction without Labeled Data. In *Proceedings of the 2009 Joint Conference of the the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing* (ACL-IJCNLP). Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/1690219.1690287

Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; Krishnamurthy, J.; Lao, N.; Mazaitis, K.; Mohamed, T.; Nakashole, N.; Platanios, E.; Ritter, A.; Samadi, M.; Settles, B.; Wang, R.; Wijaya, D.; Gupta, A.; Chen, X.; Saparov, A.; Greaves, M.; and Welling, J. 2015. Never-Ending Learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (AAAI'15). Palo Alto, CA: AAAI Press.

Nigam, K., and Ghani, R. 2000. Analyzing the Effectiveness and Applicability of Co-Training. In *Proceedings of the Ninth International Conference on Information and Knowledge Management* (CIKM'00), 86–93. New York: Association for Computing Machinery. doi.org/10.1145/354756.354805

Patwardhan, S., and Riloff, E. 2007. Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions. In *Proceedings of 2007 the Conference on Empirical Methods in Natural Language Processing* (EMNLP 2007). Stroudsburg, PA: Association for Computational Linguistics.

Phillips, W., and Riloff, E. 2002. Exploiting Strong Syntactic Heuristics and Co-Training to Learn Semantic Lexicons. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2002), 125–132. Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/1118693.1118710

Phillips, W., and Riloff, E. 2007. Exploiting Role-Identifying Nouns and Expressions for Information Extraction. Paper presented at the International Conference on Recent Advances in Natural Language Processing (RANLP'07), Borovets, Bulgaria, September 27–29.

Riloff, E. 1993. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of the 11th National Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press.

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

Riloff, E., and Jones, R. 1999. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. In

Proceedings of the Sixteenth National Conference on Artificial Intelligence. Menlo Park, CA: AAAI Press.

Riloff, E.; Qadir, A.; Surve, P.; De Silva, L.; Gilbert, N.; and Huang, R. 2013. Sarcasm as Contrast between a Positive Sentiment and Negative Situation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (EMNLP 2013). Stroudsburg, PA: Association for Computational Linguistics.

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. Stroudsburg, PA: Association for Computational Linguistics.

Riloff, E., and Wiebe, J. 2003. Learning Extraction Patterns for Subjective Expressions. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/1119355.1119369

Riloff, E.; Wiebe, J.; and Wilson, T. 2003. Learning Subjective Nouns using Extraction Pattern Bootstrapping. In *Proceedings of the Seventh Conference on Natural Language Learning* (CoNLL 2003), 25–32. Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/1119176.1119180

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. Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/980691.980751

Soderland, S. 1999. Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning* 34 (1–3): 233–272. doi.org/10.1023/A:1007562322031

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. San Francisco, CA: Morgan Kaufmann Publishers.

Stevenson, M., and Greenwood, M. 2005. A Semantic Approach to IE Pattern Induction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics,* 379–386. Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/1219840.1219887

Thelen, M., and Riloff, E. 2002. A Bootstrapping Method for Learning Semantic Lexicons Using Extraction Pattern Contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2002), 214–221. Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/1118693.1118721

Vyas, V., and Pantel, P. 2009. Semi-automatic Entity Set Refinement. In *Proceedings of North American Association for Computational Linguistics / Human Language Technology* (NAACL/HLT-09). Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/1620754.1620796

Wiebe, J., and Riloff, E. 2005. Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. In *Computational Linguistics and Intelligent Text Processing, 6th International Conference* (CICLing 2005). Lecture Notes in Computer Science 3406, 486–497. Berlin: Springer. doi.org/10.1007/978-3-540-30586-6_53

Wilson, T.; Hoffmann, P.; Somasundaran, S.; Kessler, J.; Wiebe, J.; Choi, Y.; Cardie, C.; Riloff, E.; and Patwardhan, S. 2005. OpinionFinder: A System for Subjectivity Analysis. Paper presented at the HLT/EMNLP 2005 Interactive Demonstrations, Vancouver, British Columbia, Canada, October 6–8.

Yangarber, R. 2003. Counter-Training in the Discovery of Semantic Patterns. In *Proceedings of the 41th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA: Association for Computational Linguistics.

Yangarber, R.; Grishman, R.; Tapanainen, P.; and Huttunen, S. 2000. Automatic Acquisition of Domain Knowledge for Information Extraction. In *Proceedings of the Eighteenth International Conference on Computational Linguistics* (COLING 2000). Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/992730.992782

Yarowsky, D. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics.* Stroudsburg, PA: Association for Computational Linguistics. doi.org/10.3115/981658.981684

Ellen Riloff is a professor of computer science in the School of Computing at the University of Utah. Her research area is natural language processing, focusing primarily on information extraction, semantic class induction, sentiment and affective language understanding, social media text analysis, and coreference resolution. A major emphasis of her research has been on developing bootstrapped learning methods to automatically acquire the knowledge needed for natural language processing.

Rosie Jones is a senior scientist with over 20 years of language technologies and machine learning experience. She has developed practical language and machine learning technologies in industry, through a career at Yahoo!, Akamai, and Microsoft. She helped pioneer internet industry research in web search query log analysis and user behavior analysis. She has extensive experience developing and working on research programs in industry and conducting technical transfer as demonstrated by her awarded patents, academic publications, and major industry product contributions.