Finding Mutual Benefit between Subjectivity Analysis and Information Extraction

Janyce Wiebe and Ellen Riloff

Abstract—"Subjectivity analysis" systems automatically identify and extract information relating to attitudes, opinions, and sentiments from text. As more and more people make their opinions available on the Internet and as people increasingly consult the Internet to ascertain other people's opinions about products, political issues, and so on, the demand for effective subjectivity analysis systems continues to grow. *Information extraction* systems, which automatically identify and extract factual information relating to events of interest, remain critically important in this day and age of increasingly vast amounts of text available online. In this work, we discover that these research areas are mutually beneficial. Information extraction techniques may be used to learn informative clues of subjectivity. Then, by bootstrapping from a lexicon of subjectivity clues, we can build a subjective-objective sentence classifier that does not require annotated data as input. This classifier may then be used to improve information extraction performance, on data which have not been annotated for subjectivity, by improving precision.

Index Terms—Natural language processing, text analysis.

1 Introduction

Subjectivity analysis is a rapidly growing area that involves the automatic identification and extraction of information relating to attitudes, opinions, and sentiments from unstructured text. Many applications could benefit from subjectivity analysis, including entertainment and product review mining (e.g., [1], [2], [3], [4], [5]), product reputation analysis (e.g., [5], [4], [6]), spam filtering [7], tracking sentiments toward events (e.g., [8], [9]), and multi-document summarization and question answering [10].

In contrast, *Information Extraction (IE)* systems typically involve the automatic identification and extraction of factual information relating to events. For example, IE systems have been built to extract facts associated with terrorist incidents (e.g., [11], [12], [13], [14], [15], [16]), disease outbreaks (e.g., [11], [12], [17]), plane crashes [18], vehicle launches [18], management succession [19], joint ventures [20], corporate acquisitions (e.g., [21], [22]), and job and seminar announcements (e.g., [23], [24], [22]).

Our work explores ways in which these seemingly disparate areas can benefit one another. The first research direction investigates several ways that information extraction techniques can be used to learn and recognize subjective language. Subjective language is often colorful and creative, and there is a seemingly endless variety of terms and expressions that may be used to convey opinions and emotions. Consequently, even though several subjectivity lexicons have been compiled (e.g., [25], [26], [27], [28], [29]), they are far from complete. Since subjective language is so varied, a central focus of our work is creating subjective language learners that can be trained from unannotated corpora. These learners can then be applied to large text collections to generate more expansive dictionaries, as well as small domain-specific corpora that have specialized vocabularies and idiosyncratic language. Our approach uses weakly supervised IE learning methods to automatically generate lists of subjective terms and expressions from unannotated texts. This work focuses on two types of subjective language: nouns that have a subjective meaning or connotation, and multiword expressions that capture subjectivity.

The second research direction explores the other side of the synergy: the use of subjectivity analysis to improve the accuracy of fact-based information extraction systems. Typically, IE systems look exhaustively through texts for any information that appears relevant to the domain of interest. As a result, IE systems often generate false hits from subjective sentences. One reason is that nonliteral sentences are sources of false hits for IE systems because subjective language often includes metaphorical expressions. For example, an IE system searching for bombings might incorrectly interpret the sentence *The Parliament exploded into fury* to be about a physical explosion. Other sources of false hits for IE systems are sentences with opinions, allegations, conjectures, and speculations, all of which are subjective sentences. For example, the following is an opinion sentence from which an IE system is likely to extract "the economy" as a physical target: *The subversives must suspend their efforts to destroy the economy*. We explore the idea of using a subjective sentence classifier to proactively identify and filter subjective sentences before extracting information from them. To accomplish this, we present a method for automatically creating a subjective sentence classifier for any domain, using only unannotated texts for training.

- J. Wiebe is with the Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15211. E-mail: janycewiebe@gmail.com.
- E. Riloff is with the Department of Computer Science, School of Computing, University of Utah, 50 S. Central Campus Drive, Rm 3190 MEB, Salt Lake City, UT 84112-9205.

Manuscript received 28 July 2010; revised 4 Jan. 2011; accepted 18 May 2011; published online 12 July 2011.
Recommended for acceptance by M. Walker.
For information on obtaining reprints of this article, please send e-mail to: taffc@computer.org, and reference IEEECS Log Number TAFFC-2010-07-0056.
Digital Object Identifier no. 10.1109/T-AFFC.2011.19.

This paper integrates and expands upon several different pieces of work published over a span of three years (e.g., [30], [31], [28], [32]), showing how seemingly distinct research results are ultimately integrated together for mutual benefit. In addition, this paper presents a uniform evaluation of the different subjectivity algorithms that we developed. The previous papers were published while the subjectivity corpus used for evaluation in this work, the *Multiperspective Question Answering (MPQA)* corpus [33] (described below in Section 3.1), was still evolving, so prior evaluations were performed on preliminary versions of the corpus. A final, stable version of the MPQA corpus has since been released and it has become a widely used publicly available resource for subjectivity analysis. This paper includes updated results of our algorithms on the official release of the MPQA corpus, allowing for consistent and reproducible results across systems.

1.1 Using IE to Improve Subjectivity Analysis

Our research investigates the use of weakly supervised IE learning methods to automatically acquire lists of subjective nouns and multiword subjective expressions. To learn subjective nouns, we explore two IE bootstrapping techniques, Basilisk [34] and Meta-Bootstrapping [35], which were originally designed for semantic lexicon induction. These bootstrapping algorithms begin with a few "seed nouns" that are used to identify additional nouns that occur in the same extraction pattern contexts. This approach is based on our observation that local context can select for subjective noun phrases (NPs). For example, lexico-syntactic patterns such as *"voiced ⟨np⟩"* and *"expressed ⟨np⟩"* tend to identify noun phrases that represent opinions (e.g., *"voiced displeasure"* or *"expressed disgust"*).

To learn multiword subjective expressions, we investigate the use of extraction pattern learning techniques. The main contribution of this work is the use of extraction patterns as a representation for complex subjective expressions. Subjective language often occurs in metaphorical and idiomatic expressions that cannot be adequately modeled as fixed word sequences. For example, consider the phrase "dealt a blow," which has a strong negative connotation. This phrase cannot be captured by a fixed word sequence (N-gram) because adjectives often appear inside the phrase (e.g., *"a serious blow," "a critical blow," and "a long-lasting blow"*).

Syntactically flexible expressions can be represented naturally as lexico-syntactic patterns. We use the AutoSlog-TS learner [14] to automatically identify extraction patterns that are correlated with subjective text. AutoSlog-TS does not require annotated training data, but it does require relevant and irrelevant text samples for training (in this case, subjective and objective text). This requirement is problematic for two reasons. First, ultimately our goal is to use subjectivity analysis to improve information extraction systems, which are usually domain specific. Domain-specific corpora rarely have subjectivity classifications, so manual labeling would be necessary for each domain. Second, most documents contain a mixture of subjective and objective sentences: Wiebe et al. [36], [37] report that 44 percent of sentences in their news corpus are subjective. Consequently, relying on document-level subjectivity labels during training would be noisy.

Our solution is to develop a method to automatically harvest subjective and objective sentences from unannotated texts. First, we exploit an existing subjectivity lexicon to construct domain-independent rule-based classifiers that identify subjective and objective sentences with high precision (but low recall). Then, we run the rule-based classifiers over a large, unannotated text corpus, which automatically generates a small but high-quality collection of subjective and objective sentences. These sentences are given to AutoSlog-TS as training data to learn IE patterns associated with subjectivity.

The effectiveness of the algorithms for learning subjective nouns and multiword subjective expressions is evaluated against the MPQA corpus. We also conducted a manual evaluation of the subjective nouns learned by our systems.

1.2 Using Subjectivity Analysis to Improve IE

Our second goal is to use subjectivity analysis to improve the performance of information extraction systems. Our hypothesis was that filtering subjective sentences before extracting information from them could prevent many incorrect extractions. Consequently, we need a classifier that can distinguish subjective sentences from objective (factual) sentences.

To train such a classifier, ideally we would like to have subjective and objective sentences from the relevant domain. We use the same rule-based classifiers mentioned previously to automatically harvest subjective and objective sentences from unannotated, domain-specific texts. The harvested sentences are given to a naive Bayes classifier as training data, and we embed the classifier in a self-training loop to improve its recall. The classifier uses a variety of subjectivity features, including words obtained from existing subjectivity lexicons as well as the subjective nouns and multiword expressions learned by our IE-based learning methods. Using this process, we can train a subjective sentence classifier from scratch for any unannotated corpus, including domain-specific or specialized corpora.

Finally, we train a subjective sentence classifier for the MUC-4 IE terrorism domain and experiment with several filtering strategies, including an aggressive strategy that discards all extractions from subjective sentences and more complex strategies that selectively discard extractions. We evaluate the performance of these different approaches on the MUC-4 IE data set. We find that indiscriminately filtering extractions from subjective sentences is overly aggressive, but selective filtering strategies do improve precision with minimal recall loss.

1.3 Outline

In the rest of this paper, we discuss each of these research problems, describe our methods, and present experimental results and analyses. In Section 2, we begin by positioning our work with respect to previous work on subjectivity analysis and information extraction. In Section 3, we present the weakly supervised IE-based methods for learning subjective nouns and multiword subjective expressions. In Section 4, we describe the self-trained naive Bayes classifier that learns to identify subjective and objective sentences. In Section 5, we discuss the strategies that we use to incorporate subjective sentence classification into an information extraction system and present empirical results on the MUC-4 IE task. Finally, Section 6 concludes with a summary of our results and contributions.

2 Background and Related Work

2.1 Subjectivity Analysis

Subjective expressions are words and phrases being used to express opinions, sentiments, speculations, etc. Following are some examples, with subjective expressions in bold [38]:

His **alarm** grew.
He **absorbed** the information quickly.
UCC/Disciples leaders **roundly condemned** the Iranian President's **verbal assault** on Israel.
What's the catch?

Polarity (also called *semantic orientation*) is also important for NLP applications in sentiment analysis and opinion extraction. In review mining, for example, we want to know whether an opinion about a product is positive or negative. Even so, we believe there are strong motivations for a separate subjective/objective (S/O) classification as well. First, expressions may be subjective but not have any particular polarity. An example given in [39] is *Jerome says the hospital* **feels** *no different than a hospital in the states.* An NLP application system may want to find a wide range of private states attributed to a person, such as their motivations, thoughts, and speculations, in addition to their positive and negative sentiments.

Second, distinguishing subjective and objective instances has often proven more difficult than subsequent polarity classification. Researchers have found this at various levels of analysis, including the manual annotation of phrases [40], sentiment classification of phrases [39], sentiment tagging of words [41], and sentiment tagging of word senses [42]. Thus, effective methods for S/O classification promise to improve performance for sentiment analysis. Researchers in sentiment analysis have realized benefits by decomposing the problem into S/O and polarity classification [10], [43], [39], [44]. One reason is that different features may be relevant for the two subproblems. For example, negation features are more important for polarity classification than for subjectivity classification. The overall framework we envision is a layered approach: classifying instances as objective or subjective, and further classifying the subjective instances by polarity. The current paper addresses the first of these subproblems. However, since the subproblems are interrelated, we consider both when discussing previous work below.

As mentioned above, in this paper we propose methods for learning two types of subjective language: nouns that have subjective meanings, and expressions that capture subjectivity, represented as lexico-syntactic patterns. We also develop a sentence classifier for use in IE experiments. Overall, the required inputs for our work on subjectivity are an existing subjectivity lexicon, a set of seed nouns, and a small amount of human review. Thus, our methods may be applied to new domains without requiring time-consuming manual annotations of corpora.

Several methods have been proposed for learning subjective words (also known as words with *semantic orientation*, *opinion bearing* words, *sentiment bearing* words, and so on). The original seed of our idea for learning subjective nouns, in which linguistic contexts are identified which select for subjective nouns, is due to Hatzivassiloglou and McKeown [25]. They exploit linguistic constraints on the semantic orientations of adjectives used in conjunctions: In the construct *<adjective> and <adjective>*, the adjectives must be the same polarity. We can view the pattern *<adjective-known-to-be-positive> and <adjective>* as an extraction pattern for finding positive adjectives (similarly for negative ones). This idea was later extended by several other researchers, including Turney [45], who uses five patterns to extract positive and negative words, Gamon and Aue [46], who exploit the constraint that words of opposite polarity tend not to occur in the same sentence, and Kanayama and Nasukawa [47], who exploit the tendency for the same polarities to appear successively in context. However, such patterns and linguistic constraints used by other researchers are manually conceived, while we identify the extraction patterns automatically. Further, our system takes into account a greater variety of local contexts to identify subjective words than in others' work, due to our use of extraction patterns, which require shallow parsing and syntactic role assignment.

Turning to lexico-syntactic representations of subjective expressions, along with Yi et al. in 2003 [6], we were the first to automatically identify subjective clues that are lexico-syntactic patterns rather than individual words or n-grams (the year before, Pang et al. [2] and Wiebe and Wilson [48] experimented with n-grams). Since then, pattern representations of subjective expressions have been more common (e.g., [47], [29], [49]). Again, our system learns a wider variety of patterns than in other work. Note that some researchers use patterns to link opinions to their sources (holders) and/or targets (topics) (e.g., [50], [51], [29]), a task that we do not address in this work.

Our sentence classifier is a traditional machine learning system. The interesting aspect of the classifier is that it is trained on data that are created automatically, not on manually annotated data. Yet, the performance rivals the performance of classifiers trained on manually annotated data [30], [28] (see Section 4).

There is other work in opinion mining and subjectivity and sentiment analysis that does not require manually labeled data as input, including [45], [52], [47], [40], [53]. We were the first to exploit an existing lexical resource to mine unlabeled data for subjective and objective sentences, to serve as training data for subjectivity learners.

Recently, domain dependence in opinion mining and sentiment analysis has been addressed by several groups. Blitzer et al. [54] focus on domain adaptation via structural correspondence learning and Andreevskaia and Bergler [55] combine a supervised classifier with a system that mines WordNet synsets and glosses for relevant knowledge. Those are both different strategies from ours of bootstrapping from an existing subjectivity lexicon to classify data directly in the domain of interest. Kanayama and Nasukawa's [47] work is the most similar recent work. They exploit an existing domain-independent lexicon to find domain-dependent clues in a corpus, with mechanisms added to distinguish among positive, negative, and neutral clues.

Note that our strategy is different from the strategy of using "found" training data, such as reviews with user-defined stars and movie-review snippets [43]. With ours, the data are not limited to the domains of the found data.

2.2 Information Extraction

Our research revolves around event-oriented information extraction, where the goal of the IE system is to extract facts associated with domain-specific events from unstructured text. Information extraction systems have been created for a variety of domains, including terrorism [16], [13], [14], [15], plane crashes [18], vehicle launches [18], management succession [19], joint ventures [20], corporate acquisitions [21], [22], job postings [23], [22], seminar announcements [24], [22], and disease outbreaks [12], [56], [17].

Many different approaches have been developed, but generally speaking, they fall into two categories: classifier-based approaches and rule/pattern-based approaches. *Classifier-based IE systems* use machine learning techniques to train a classifier that sequentially processes a document looking for words to be extracted. Examples of classifier-based IE systems are SRV [21], HMM approaches [22], ALICE [13], and Relational Markov Networks [57]. The classifier decides whether a word should be extracted by considering features associated with that word as well as features of the words around it. *Pattern-based IE systems* use a set of explicit patterns or rules to find relevant information. Some older systems relied on hand-crafted patterns, while more recent systems learn them automatically or semi-automatically. Examples of rule/pattern-based approaches to information extraction are PALKA [58], LIEP [59], CRYSTAL [15], AutoSlog/AutoSlog-TS [60], [14], RAPIER [61], ExDisco [19], SNOWBALL [62], $(LP)^2$ [24], subtree patterns [63], predicate-argument rules [64], and KnowItAll [65].

Our research utilizes pattern-based IE methods, both for learning subjective expressions and for incorporating subjectivity analysis into IE applications. As we will explain in Section 3.3, the pattern representations used for IE naturally lend themselves to a flexible representation for subjective expressions. The pattern-based bootstrapping methods described in Section 3.4 also provide a natural mechanism for learning subjective words.

Our work is the first research effort to exploit subjectivity analysis to improve the performance of an information extraction system (see [32] for our earliest results of this effort). Along similar lines, Klebanov et al. [66] developed a method for simplifying natural language texts to make them easier to process by information-seeking applications. The relation to our work is that, as part of their process, they filter out sentences with verbs such as "want" and "desire" because they are not factive (e.g., from "John wants to win," we infer that he has not already won). Thus, their system filters out some subjective sentences. However, they do not experiment with using the results of their simplification algorithm to improve the performance of an end application, stating that the performance of the algorithm is not yet satisfactory.

3 Learning Subjective Words and Expressions with Extraction Patterns

This section presents our work exploiting IE-based methods to learn subjective expressions, both learning subjective nouns and learning multiword subjective expressions. For the first task, extraction patterns are exploited to identify local contexts which may select for subjective nouns. The outcome of this task is a list of nouns with subjective usages. For the second task, extraction patterns are themselves the goal. The outcome is a list of extraction patterns correlated with subjectivity in a corpus.

3.1 Data

For the experiments described in this section and in Section 4, we require unannotated data for input, as well as annotated data for evaluation. We use collections of English-language versions of news articles from the world press made available by FBIS, the US Foreign Broadcast Information Service. The annotated subset of these data is the Multi-perspective Question Answering Corpus, which has been annotated for subjective expressions.[1] In this section and in Section 4, all of the gold standards are sentence-level subjectivity classifications. The sentence-level classes are defined in terms of the expression level annotations as follows: A sentence is subjective if it contains one or more subjective expressions of medium or higher intensity.

3.2 Extraction Patterns

Our research uses *extraction patterns* to represent linguistic expressions, both for generating lists of subjective nouns and as features for subjective/objective sentence classification. In this section, we briefly describe the extraction pattern representation that we use in all of our algorithms.

The extraction patterns are lexico-syntactic patterns that represent one or more words appearing in a specific syntactic context. A shallow parser, Sundance [67], produces a syntactic analysis of each sentence before the patterns are applied. The Sundance parser identifies major syntactic constituents (e.g., NPs, VPs, and PPs), labels VPs with respect to voice (active voice, passive voice, or infinitive), segments each sentence into clauses, and identifies the syntactic Subjects, Direct Objects, and Indirect Objects in each clause.

Fig. 1 shows the 13 syntactic templates for the extraction patterns used in this work. The patterns extract noun phrases from three syntactic positions: Subjects (*subj*), Direct Objects (*dobj*), and within Prepositional Phrases (*prep np*). The brackets (< >) in Fig. 1 indicate the syntactic constituent that is extracted by the pattern. An extraction pattern is created by instantiating one of the syntactic templates with specific words. Each word must match the head of the corresponding constituent (e.g., a noun must match the head of an NP). For example, the first template shown in Fig. 1 could be instantiated with different verbs to create patterns that match instances of those verbs that occur in passive voice VP constructions. For example, the pattern **<subj> passive-verb(*kidnapped*)** would match sentences such as "John was brazenly kidnapped" or "Two

1. The MPQA corpus is described in [33] and is available at www.cs.pitt.edu/mpqa.

<subj> passive-verb
<subj> active-verb
<subj> active-verb dobj
<subj> verb infinitive
<subj> aux noun
active-verb <dobj>
infinitive <dobj>
verb infinitive <dobj>
noun aux <dobj>
noun prep <np>
active-verb prep <np>
passive-verb prep <np>
infinitive prep <np>

Fig. 1. Syntactic templates for extraction patterns.

reporters in Cancun were kidnapped." The syntactic subject of the VP ("John" and "Two reporters" in the previous sentences) will be extracted. It is important to note that the pattern words and the extracted NP do not have to be adjacent; they only need to occur in the appropriate syntactic constructions.

Intuitively, each pattern matches words in the context surrounding a noun phrase that reveal the role that the NP plays with respect to an action or concept. For example, the pattern **<subj> active-verb(*attacked*)** will extract NPs that are acting as the attackers, while the pattern **active-verb(*attacked*) <dobj>** will extract NPs that are the object of the attack. Some of the patterns capture noun relations, such as **noun(*attack*) prep(*of*) <np>**. The syntactic templates containing *aux* can be instantiated with either "to-be" or "to-have" as auxiliary verbs to capture predicate nominal constructions or "have" expressions, respectively. For example, the pattern **noun(*president*) aux(*to-be*) <dobj>** would match "The U.S. president is Barack Obama."

The algorithms described in this paper generate extraction patterns automatically from a text corpus by instantiating the syntactic templates exhaustively. All of the syntactic templates are matched against every sentence, and each matching template produces a lexico-syntactic pattern with the corresponding words from the sentence. For example, consider the sentence:

"She wanted desperately to believe in humanity."

Exhaustively matching the syntactic templates against this sentence would produce four patterns:

<subj> active-verb(*wanted*)
<subj> verb(*wanted*) infinitive(*believe*)
infinitive(*believe*) <dobj>
verb(*wanted*) infinitive(*believe*) <dobj>

For the sake of simplicity, in the remainder of this paper, we will generally show a pattern using just the general expression that it would match (e.g., *wanted to believe*).

3.3 Learning Subjective Nouns Using Extraction Pattern Contexts

Two bootstrapping algorithms have been developed to create semantic dictionaries by exploiting extraction patterns: Meta-Bootstrapping [35] and Basilisk [34].

Meta-Bootstrapping and Basilisk were designed to learn words that belong to a semantic category (e.g., "truck" is a VEHICLE and "seashore" is a LOCATION). Both algorithms begin with unannotated texts and *seed* words that represent a semantic category. A bootstrapping process looks for words that appear in the same extraction patterns as the seeds and hypothesizes that those words belong to the same semantic class. The principle behind this approach is that words of the same semantic class appear in similar pattern contexts. For example, the phrases "lived in" and "traveled to" will co-occur with many noun phrases that represent LOCATIONS.

In our research, we want to automatically identify words that are subjective. Subjective terms have many different semantic meanings, but we believe that the same contextual principle applies to subjectivity. In this section, we overview these bootstrapping algorithms and explain how we use them to generate lists of subjective nouns.

3.3.1 Meta-Bootstrapping

The Meta-Bootstrapping ("MetaBoot") process [35] begins with a small set of seed words that represent a targeted semantic category (e.g., 10 words that represent LOCATIONS) and an unannotated corpus. First, MetaBoot automatically creates a set of extraction patterns for the corpus by applying and instantiating syntactic templates. This process literally produces thousands of extraction patterns that, collectively, will extract every noun phrase in the corpus. Next, MetaBoot computes a score for each pattern based upon the number of seed words among its extractions. The best pattern is saved and *all* of its extracted noun phrases are automatically labeled as the targeted semantic category.[2] MetaBoot then rescores the extraction patterns, using the original seed words as well as the newly labeled words, and the process repeats. This procedure is called *mutual bootstrapping*.

A second level of bootstrapping (the "meta" bootstrapping part) makes the algorithm more robust. When the mutual bootstrapping process is finished, all nouns that were put into the semantic dictionary are reevaluated. Each noun is assigned a score based on how many different patterns extracted it. Only the five best nouns are allowed to remain in the dictionary. The other entries are discarded, and the mutual bootstrapping process starts over again using the revised semantic dictionary.

3.3.2 Basilisk

Basilisk [34] is a more recent bootstrapping algorithm that also utilizes extraction patterns to create a semantic dictionary. Similarly, Basilisk begins with an unannotated text corpus and a small set of seed words for a semantic category. The bootstrapping process involves three steps. 1) Basilisk automatically generates a set of extraction patterns for the corpus and scores each pattern based upon the number of seed words among its extractions. This step is identical to the first step of Meta-Bootstrapping. Basilisk then puts the best patterns into a Pattern Pool. 2) All nouns[3] extracted by a pattern in the Pattern Pool are put into a Candidate Word Pool. Basilisk scores each noun based upon the set of patterns that extracted it and their collective association with the seed words. 3) The top 10 nouns are labeled as the targeted semantic class and are added to the

2. Our implementation of Meta-Bootstrapping learns individual nouns (versus noun phrases) and discards capitalized words.

3. Technically, each head noun of an extracted noun phrase.

TABLE 1
Subjective Seed Words from FBIS

cowardice	embarrassment	hatred	outrage
crap	fool	hell	slander
delight	gloom	hypocrisy	sigh
disdain	grievance	love	twit
dismay	happiness	nonsense	virtue

dictionary. The bootstrapping process then repeats, using the original seeds and the newly labeled words.

The main difference between Basilisk and Meta-Bootstrapping is that Basilisk scores each noun based on *collective* information gathered from *all* patterns that extracted it. In contrast, Meta-Bootstrapping identifies a single best pattern and assumes that everything it extracted belongs to the same semantic class. The second level of bootstrapping smooths over some of the problems caused by this assumption. In comparative experiments [34], Basilisk outperformed Meta-Bootstrapping. But since our goal of learning subjective nouns is different from the original intent of the algorithms, we tried them both. We also suspected that they might learn different words, in which case using both algorithms could be worthwhile.

3.3.3 Application to Learning Subjective Nouns

The Meta-Bootstrapping and Basilisk algorithms need seed words and an unannotated text corpus as input. Since we did not need annotated texts, we created a much larger training corpus, the *bootstrapping corpus*, by gathering 950 new texts from the FBIS source mentioned in Section 3.1. To find candidate seed words, we automatically identified 850 nouns that are positively correlated with subjective sentences in another data set. However, it is crucial that the seed words occur frequently in our unannotated texts or the bootstrapping process will not get off the ground. So, we searched for each of the 850 nouns in the bootstrapping corpus, sorted them by frequency, and manually selected 20 high-frequency words that we judged to be strongly subjective. Table 1 shows the 20 seed words used for both Meta-Bootstrapping and Basilisk.

We ran each bootstrapping algorithm for 400 iterations, generating five words per iteration. Basilisk generated 2,000 nouns and Meta-Bootstrapping generated 1,996 nouns.[4] Table 2 shows some examples of extraction patterns that were discovered to be associated with subjective nouns. The extraction pattern is shown on the left, and subjective words extracted by the pattern in our unannotated data set are shown on the right.

Meta-Bootstrapping and Basilisk are semi-automatic lexicon generation tools because, although the bootstrapping process is 100 percent automatic, the resulting lexicons need to be reviewed by a human. This is because NLP systems expect dictionaries to have high integrity. Even if the algorithms could achieve 90 percent accuracy, a dictionary in which one of every 10 words is defined incorrectly would probably not be desirable. So, we manually reviewed the 3,996 words proposed by the algorithms. This process is very fast; it takes only a few seconds to classify each word. The entire review process took approximately 3-4 hours. One author did this labeling; this person had not looked at or run tests on the MPQA corpus used for evaluation below.

We classified the words as *StrongSubjective*, *WeakSubjective*, or *Objective*. *Objective* terms are not subjective at all (e.g., "chair" or "city"). *StrongSubjective* terms have strong subjective connotations, such as "bully" or "belligerence." *WeakSubjective* is used for three situations: 1) Words that have weak subjective connotations, such as "aberration," which implies something out of the ordinary but does not evoke a strong sense of judgment. 2) Words that clearly have multiple senses or uses, where one is subjective but the other is not.[5] For example, the word "plague" may refer to a disease (objective) or an onslaught of something negative (subjective). 3) Words that are objective by themselves but appear in idiomatic expressions that are subjective. For example, the word "eyebrows" is labeled *WeakSubjective* because the expression "raised eyebrows" probably occurs more often in our corpus than literal references to "eyebrows." Table 3 shows examples of learned words that we classified as *StrongSubjective* or *WeakSubjective*.

TABLE 2
Extraction Pattern Examples

Extraction Patterns	Examples of Extracted Nouns
expressed <dobj>	condolences, hope, grief, views, worries, recognition
indicative of <np>	compromise, desire, thinking
inject <dobj>	vitality, hatred
reaffirmed <dobj>	resolve, position, commitment
voiced <dobj>	outrage, support, skepticism, disagreement, opposition, concerns, gratitude, indignation
show of <np>	support, strength, goodwill, solidarity, feeling
<subject> was shared	anxiety, view, niceties, feeling

TABLE 3
Examples of Learned Subjective Nouns

Strong Subjective		Weak Subjective	
tyranny	scum	aberration	plague
smokescreen	bully	allusion	risk
apologist	devil	apprehensions	drama
barbarian	liar	beneficiary	trick
belligerence	pariah	resistant	promise
condemnation	venom	credence	intrigue
sanctimonious	diatribe	distortion	unity
exaggeration	mockery	eyebrows	failures
repudiation	anguish	inclination	tolerance
insinuation	fallacies	liability	persistent
antagonism	evil	assault	trust
atrocities	genius	benefit	success
denunciation	goodwill	blood	spirit
exploitation	injustice	controversy	slump
humiliation	innuendo	likelihood	sincerity
ill-treatment	revenge	peaceful	eternity
sympathy	rogue	pressure	rejection

4. Meta-Bootstrapping will sometimes produce fewer than five words per iteration if it has low confidence in its judgments.

5. In this work, the judgment that a word has both subjective and objective senses was an intuitive judgment, without reference to any particular dictionary. Recent work addresses the issue of subjective and objective word senses more formally, e.g., [38], [42].

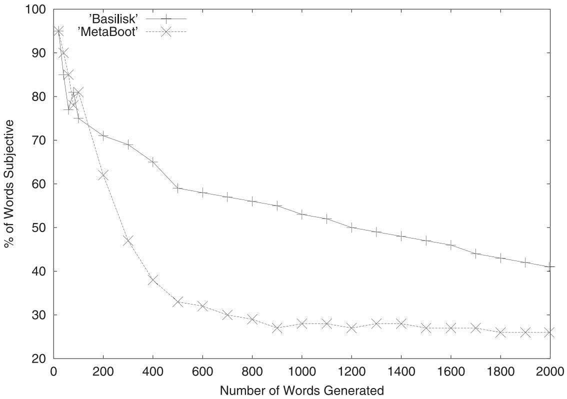

Fig. 2. Accuracy during bootstrapping.

Once the words are manually classified, we can measure the effectiveness of the algorithms. We do this in two ways in this section.

First, we track the accuracy of the words as the bootstrapping process progresses. The graph in Fig. 2 shows these results. The X-axis shows the number of words generated so far. The Y-axis shows the percentage of those words that are manually classified as subjective. As is typical of bootstrapping algorithms, accuracy is high during the initial iterations but tapers off as bootstrapping continues. After 20 words, both algorithms are 95 percent accurate. After 100 words, Basilisk is 75 percent accurate and MetaBoot is 81 percent accurate. After 1,000 words, accuracy drops to about 28 percent for MetaBoot, but Basilisk is still performing reasonably well at 53 percent. Although 53 percent accuracy is not high for a fully automatic process, Basilisk depends on a human to review the words, so 53 percent accuracy means that the human is accepting every other word, on average. Thus, the reviewer's time is still being spent productively even after 1,000 words have been hypothesized. Table 4 shows the size of the final lexicons created by the bootstrapping algorithms. The first two columns show the number of subjective terms learned by Basilisk and Meta-Bootstrapping. Basilisk is more prolific, generating 825 subjective terms, compared to 522 for Meta-Bootstrapping. The third column shows the intersection between their word lists. There is substantial overlap, but both algorithms produce many words that the other does not. The last column shows the results of merging their lists. In total, the bootstrapping algorithms produced 1,052 subjective nouns.

The second way we measure the effectiveness of the algorithms is to evaluate the nouns against the MPQA corpus, which, recall, is manually annotated for subjectivity. In particular, we want to determine if the nouns consistently occur in subjective sentences. We evaluate a simple classifier consisting of one rule: If a sentence contains one of the nouns, the sentence is subjective; otherwise, the sentence is objective. Of course, we do not intend the nouns to be used by themselves for subjectivity classification, but rather integrated with other subjectivity clues and features (as in Sections 3.4.1 and 4 below). However, evaluating the simple classifier is a transparent way to evaluate the nouns.

TABLE 4
Subjective Word Lexicons after Manual Review
(B = Basilisk, M = MetaBootstrapping)

	B	M	$B \cap M$	$B \cup M$
StrongSubj	372	192	110	454
WeakSubj	453	330	185	598
Total	825	522	295	1052

TABLE 5
Evaluation of Nouns against a Gold Standard

	SubjRecall	SubjPrec
Basilisk StrongSubj	2.4	86.8
Basilisk WeakSubj	4.6	75.9
MetaBoot StrongSubj	13.9	83.1
MetaBoot WeakSubj	36.9	72.6

We evaluate this classifier with respect to *subjective precision* (*SubjPrec*), the percentage of sentences automatically classified as subjective that are truly subjective (i.e., the conditional probability that a sentence truly is subjective, given that it is automatically classified as subjective), and *subjective recall* (*SubjRecall*), the percentage of true subjective sentences that are automatically classified as subjective (i.e., the conditional probability that a sentence is automatically classified as subjective, given that it is truly subjective).

Table 5 shows results for the classifier using each type of noun in turn. For example, the first row shows results for the nouns identified by Basilisk which were judged to be strongly subjective, and the last row shows results for the nouns identified by Metaboot which were judged to be weakly subjective. The data set consists of 9,732 sentences, 5,380 of which are subjective (55 percent). This data set does not overlap with the (unannotated) data set used to learn the nouns.

Given that the percentage of subjective sentences in the data is only 55 percent, the precision of all four sets is high, with the strongly subjective sets having higher precision than the weakly subjective sets, as expected. MetaBoot is higher recall and lower precision than Basilisk. MetaBoot's advantage in recall is greater than its disadvantage in precision, suggesting that if one of the algorithms is to be chosen, MetaBoot may be preferable. However, as discussed above, truly comprehensive subjectivity classification will require knowledge not only of frequent reliable clues, but also of large numbers of infrequent clues. Table 4 shows that Basilisk yields more words than MetaBoot, and Table 5 shows that Basilisk has higher precision (comparing the two StrongSubj sets and the two WeakSubj sets). These findings suggest that applying both algorithms is valuable.

In [30], the effectiveness of the nouns learned with extraction pattern bootstrapping is additionally evaluated in the context of supervised learning on manually annotated data. We found that statistically significant performance improvements can be achieved by adding the learned nouns to the subjectivity lexicon (which is used by the classifiers presented in that paper), demonstrating that the learned nouns represent new knowledge not contained in the subjectivity lexicon before. In the current paper, Section 3.4.1 below evaluates our rule-based classifiers with and without the learned nouns, providing additional evidence that the nouns represent new knowledge not already contained in the subjectivity lexicon.

3.4 Learning Subjective Expressions as Extraction Patterns

We now turn to using lexico-syntactic patterns as linguistically rich and flexible representations of subjective expressions, and learning subjective patterns from unannotated data.

One of our hypotheses was that extraction patterns would be able to represent subjective expressions that have noncompositional meanings. For example, consider the common expression *drives (someone) up the wall*, which expresses the feeling of being annoyed with something. The meaning of this expression is quite different from the meanings of its individual words (*drives*, *up*, *wall*). Furthermore, this expression is not a fixed word sequence that could easily be captured by N-grams. It is a relatively flexible construction that may be more generally represented as *<x> drives <y> up the wall*, where x and y may be arbitrary noun phrases. This pattern would match many different sentences, such as "George drives me up the wall," "She drives the mayor up the wall," or "The nosy old man drives his quiet neighbors up the wall."

We also wondered whether the extraction pattern representation might reveal slight variations of the same verb or noun phrase that has different connotations. For example, you can say that *a comedian bombed last night*, which is a subjective statement, but you can't express this sentiment with the passive voice of *bombed*.

To explore these hypotheses, we adopt a learning process very similar to that used by AutoSlog-TS, which requires only relevant texts (in our case, a set of subjective sentences) and irrelevant texts (in our case, a set of objective sentences) as its input. We begin this section by describing the method used to obtained these required sets.

3.4.1 Harvesting Training Data from Unannotated Texts

This section describes our method for automatically identifying subjective and objective sentences from unannotated texts. The resulting sets of sentences are used as training data for the extraction pattern learning process described in Section 3.4.2, and to create the naive Bayes classifier (described in Section 4) which is used in the Information Extraction experiments described in Section 5.

The process begins with a large collection of unannotated text and two high-precision subjectivity classifiers. One classifier searches the unannotated corpus for sentences that can be labeled as subjective with high confidence, and the other classifier searches for sentences that can be labeled as objective with high confidence. All other sentences in the corpus are left unlabeled. The process is depicted in Fig. 3.

The high-precision classifiers (HP-Subj and HP-Obj) use lists of lexical items that have been shown in previous work to be good subjectivity clues. Most of the items are single words, some are N-grams, but none involve syntactic generalizations as in the extraction patterns. Any data used to develop this vocabulary do not overlap with any of the training or test data used in this paper.

Many of the subjectivity clues are from manually developed resources, including entries from [68], [69], Framenet lemmas with frame element *experiencer* [70], adjectives manually annotated for polarity [25], and subjectivity clues listed in [71]. Others were derived from corpora, including the subjective nouns learned from unannotated data as described above in Section 3.3.

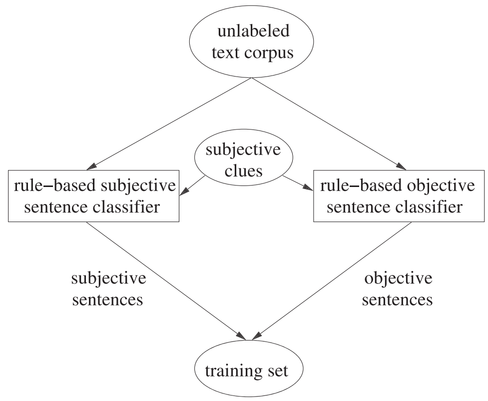

Fig. 3. Training data creation.

The subjectivity clues are divided into those that are strongly subjective and those that are weakly subjective, using a combination of manual review and empirical results on a small training set of manually annotated data. These terms have the same meanings as they did in Section 3.3: A *strongly subjective* clue is one that is seldom used without a subjective meaning, whereas a *weakly subjective* clue is one that commonly has both subjective and objective uses.

The high-precision subjective classifier classifies a sentence as subjective if it contains two or more of the strongly subjective clues. We evaluate this classifier against the same subset of the MPQA data as used in Section 3.3, which, recall, consists of 9,732 sentences, 5,380 of which are subjective (55 percent).[6]

On the manually annotated test set used in this paper, this classifier achieves 91.7 percent precision and 30.9 percent recall (that is, 91.7 percent of the sentences that it selected are subjective, and it found 30.9 percent of the subjective sentences in the test set). The subjective F-measure is 46.2 percent.

The high-precision objective classifier takes a different approach. Rather than looking for the presence of lexical items, it looks for the absence of them. It classifies a sentence as objective if there are no strongly subjective clues and at most two weakly subjective clues in the current, previous, and next sentence combined. Why doesn't the objective classifier mirror the subjective classifier and consult its own list of strongly objective clues? There are certainly lexical items that are statistically correlated with the *objective* class (examples are cardinal numbers [72] and words such as *per*, *case*, *market*, and *total*), but the presence of such clues does not readily lead to high-precision objective classification. If we add sarcasm or a negative evaluation to a sentence about a dry topic such as stock prices, then the sentence becomes subjective. Conversely, even if we add objective topics to a sentence containing two

6. The results in this paper differ somewhat from the results published in [31], [28] for two reasons: Smaller manually annotated data sets are used in the previous papers (and, in fact, those sets are different from each other), and the subjectivity lexicon used in the current paper has additional entries, derived from other research carried out since that time. However, we have the same pattern of results. That is, the same relative increases and decreases in performance are observed in the current paper as in the previous ones.

strongly subjective words such as *odious* and *scumbag*, the sentence remains subjective.

The high-precision objective classifier has higher recall and lower precision than the subjective classifier, with comparable F-measure: 83 percent precision and 32.8 percent recall on the test set mentioned above (that is, 83 percent of the sentences selected by the objective classifier are objective, and the objective classifier found 32.8 percent of the objective sentences in the test set); the objective F-measure is 47.1 percent. Though the precision of the objective classifier is lower than the precision of the subjective classifier, the performance proved to be good enough for our purposes.

Recall that we are interested in the performance of the rule-based classifiers when they do not include the nouns learned as described in Section 3.3, to assess whether the nouns represent new knowledge added to the subjectivity lexicon. When the nouns are excluded from the subjectivity lexicon,[7] subjective precision is a bit higher (by 1.6 percentage points). However, subjective recall is considerably lower (by 7.6 percentage points), showing that the nouns do represent new knowledge.

The subjective and objective classifiers mirror one another since one looks for the presence and the other the absence of subjective clues. The results reflect this: Without the nouns, objective recall increases (by 10.5 percentage points) but objective precision decreases (by 9.4 percentage points). The resulting objective precision (73.6) is too low for our purposes since these classifiers are being used to create training data.

We apply the classifiers (with the nouns included) to an unannotated corpus of 298,954 sentences from the world press (which does not overlap with the annotated data set used for evaluation). A total of 68,580 sentences are selected by the objective classifier and 48,814 are selected by the subjective classifier. Thus, close to 40 percent of the sentences are harvested from the unannotated data store.

3.4.2 Learning Subjective Extraction Patterns

To automatically learn extraction patterns that are associated with subjectivity, we use a learning algorithm similar to AutoSlog-TS [14]. For training, AutoSlog-TS uses a text corpus consisting of two distinct sets of texts: "relevant" texts (in our case, subjective sentences) and "irrelevant" texts (in our case, objective sentences). A set of syntactic templates represents the space of possible extraction patterns.

The learning process has two steps. First, the syntactic templates are applied to the training corpus in an exhaustive fashion so that extraction patterns are generated for (literally) every possible instantiation of the templates that appears in the corpus. The left column of Fig. 4 shows the syntactic templates used by AutoSlog-TS. The right column shows specific extraction patterns that are learned during our subjectivity experiments as instantiations of the syntactic forms on the left. For example, the pattern *<subj> is satisfied* will match any sentence where the verb *satisfied* appears in the passive voice. The pattern *<subj> dealt blow*

SYNTACTIC FORM	EXAMPLE PATTERN
<subj> passive-verb	<subj> was satisfied
<subj> active-verb	<subj> complained
<subj> active-verb dobj	<subj> dealt blow
<subj> verb infinitive	<subj> appear to be
<subj> aux noun	<subj> has position
active-verb <dobj>	endorsed <dobj>
infinitive <dobj>	to condemn <dobj>
verb infinitive <dobj>	get to know <dobj>
noun aux <dobj>	fact is <dobj>
noun prep <np>	opinion on <np>
active-verb prep <np>	agrees with <np>
passive-verb prep <np>	was worried about <np>
infinitive prep <np>	to resort to <np>

Fig. 4. Syntactic templates and examples of patterns that were learned.

represents a more complex expression that will match any sentence that contains a verb phrase with head = *dealt* followed by a direct object with head = *blow*. This would match sentences such as "The experience dealt a stiff blow to his pride." It is important to recognize that these patterns look for specific syntactic constructions produced by a (shallow) parser, rather than exact word sequences.

The second step of AutoSlog-TS's learning process applies all of the learned extraction patterns to the training corpus and gathers statistics for how often each pattern occurs in subjective versus objective sentences. AutoSlog-TS then ranks the extraction patterns using a metric called RlogF [14] and asks a human to review the ranked list and make the final decision about which patterns to keep.

The process used here differs in two ways. First, for this work, we want a fully automatic process that does not depend on a human reviewer. As illustrated below, patterns that are variations of each other may distribute differently in subjective versus objective texts, and it would be difficult, if not impossible, for a human to predict the differences. Thus, in contrast with the nouns learned above in Section 3.3, we do not intend the learned extraction patterns to be reviewed by a human and added to a subjectivity lexicon,[8] but rather to provide evidence of subjectivity which is combined with other evidence for classification, such as the knowledge contained in a subjectivity lexicon.

Second, we are most interested in finding patterns that can identify patterns associated with subjectivity with high precision. So, we rank the extraction patterns using a conditional probability measure: the probability that a sentence is subjective given that a specific extraction pattern appears in it. The exact formula is

$$Pr(subjective \mid pattern_i) = \frac{subjfreq(pattern_i)}{freq(pattern_i)},$$

where $subjfreq(pattern_i)$ is the frequency of $pattern_i$ in subjective training sentences, and $freq(pattern_i)$ is the frequency of $pattern_i$ in all training sentences. (This may also be viewed as the subjective precision of the pattern on the training data.) Finally, we use two thresholds to select

7. If one of the nouns was already in the subjectivity lexicon, for example, if it was added as part of a list from a manually developed resource such as [68], then it is *not* removed for the purpose of this evaluation. We are assessing what knowledge is *added* as a result of the experiments described in Section 3.3.

8. The current release of the lexicon available at http://www.cs.pitt.edu/mpqa does not include them, while it does include the nouns from Section 3.3.

PATTERN	FREQ	%SUBJ
<subj> was asked	11	100%
<subj> asked	128	63%
<subj> is talk	5	100%
talk of <np>	10	90%
<subj> will talk	28	71%
<subj> put an end	10	90%
<subj> put	187	67%
<subj> is going to be	11	82%
<subj> is going	182	67%
was expected from <np>	5	100%
<subj> was expected	45	42%
<subj> is fact	38	100%
fact is <dobj>	12	100%

Fig. 5. Patterns with interesting behavior.

extraction patterns that are strongly associated with subjectivity in the training data. For this work, we choose extraction patterns for which $freq(pattern_i) \geq \theta_1 = 5$ and $Pr(subjective \mid pattern_i) \geq \theta_2 = 0.95$.

Fig. 5 shows some patterns learned by our system, the frequency with which they occur in the training data (**FREQ**) and the percentage of times they occur in subjective sentences (**%SUBJ**). For example, the first two rows show the behavior of two similar expressions using the verb *asked*. Of the sentences that contain *asked* in the passive voice, 100 percent are subjective, but only 63 percent of the sentences that contain *asked* in the active voice are subjective. A human would probably not expect the active and passive voices to behave so differently. To understand why this is so, we looked in the training data and found that the passive voice is often used to query someone about a specific opinion. For example, here is one such sentence from our training set: "Ernest Bai Koroma of RITCORP was asked to address his supporters on his views relating to 'full blooded Temne to head APC'." In contrast, many of the sentences containing *asked* in the active voice are more general in nature, such as "The mayor asked a newly formed JR about his petition."

Fig. 5 also shows that expressions using *talk* as a noun (e.g., "Fred is the talk of the town") are highly correlated with subjective sentences, while *talk* as a verb (e.g., "The mayor will talk about...") are found in a mix of subjective and objective sentences. Not surprisingly, longer expressions tend to be more idiomatic (and subjective) than shorter expressions (e.g., *put an end (to)* versus *put*; *is going to be* versus *is going*; *was expected from* versus *was expected*). Finally, the last two rows of Fig. 5 show that expressions involving the noun *fact* are highly correlated with subjective expressions! These patterns match sentences such as *The fact is...* and ... *is a fact*, which apparently are often used in subjective contexts. This example illustrates that the corpus-based learning method can find phrases that might not seem subjective to a person intuitively, but that are reliable indicators of subjectivity.

We trained the extraction pattern learner on the training data created by the rule-based classifiers described above in Section 3.4.1. A total of 8,490 extraction patterns were learned that have $freq(pattern_i) \geq \theta_1 = 5$ and $Pr(subjective \mid pattern_i) \geq \theta_2 = 0.95$ in the training data.

We argued in Section 3.4.1 that clues associated with objectivity do not readily lend themselves to high-precision classification. However, perhaps combined with other information, learning objective extraction patterns might be useful. To explore this, we extract the extraction patterns that are negatively correlated with subjective sentences (and hence positively correlated with objective sentences), namely, those that have $freq(pattern_i) \geq \theta_1 = 5$ and $Pr(subjective \mid pattern_i) \leq \theta_2 = 0.15$ in the training data, and treat them as objective extraction patterns. A total of 2,910 extraction patterns are selected using these parameters.

As we did for the nouns in Section 3.3, we evaluate the learned patterns against manually annotated data. Specifically, we evaluate a simple classifier that classifies a sentence as subjective if it contains any of the learned subjective patterns. Evaluated on the manually annotated MPQA data set used in this paper, the subjective precision of the set of learned patterns is 80.7 (i.e., 80.7 percent of the sentences with subjective patterns are subjective) and the subjective recall is 47.2 (i.e., 47.2 percent of the subjective sentences contain at least one subjective pattern). Thus, the set of learned patterns has fairly high precision while retrieving a nontrivial proportion (close to 50 percent) of the subjective sentences in the manually annotated corpus.

Applying the same evaluation to the objective patterns, the objective precision of the set of learned patterns is 70.0 and the objective recall is 18.3. Thus, as we expected, the objective patterns (learned in this setting) are lower precision than the subjective patterns.

As shown in [31], varying the parameters θ_1 (frequency) and θ_2 (conditional probability) results in the expected trade-off between precision and recall (with subjective precision ranging from 71 to 85 percent, and subjective recall ranging from 40 to 91 percent on the data set used in that paper).

The question arises, does the system learn any additional knowledge, over and above the subjective vocabulary used by the rule-based classifiers? The answer is yes. If we add the extraction pattern features to the "strongly subjective" category of clues used by the classifiers, and use exactly the same strategy as in the original subjective rule-based classifier, the results change as follows: Subjective recall increases by 19.8 percentage points, to 50.7, while precision decreases by much less (7.3 points, to 84.4).

The objective rule-based classifier may also be (modestly) improved by incorporating the patterns as well. As for the subjective classifier, the subjective extraction patterns are added to the "strongly subjective" category of clues. The classification strategy is then augmented as follows: In addition to using its previous rules, the objective classifier also labels a sentence as objective if it contains no strongly subjective clues but at least one objective extraction pattern. Note that adding the subjective extraction patterns to the set of strongly subjective clues works to *decrease* the recall of the objective classifier because it looks for the absence of subjectivity clues. Adding the objective clues may balance that effect, serving to *increase* the recall of the objective classifier. The net result is that objective recall increases by 4.9 percentage points, to 37.7, while precision decreases by only 0.5, to 82.5.

Adding extraction patterns to the rule-based classifiers expands their coverage with relatively smaller drops in precision.

4 Learning Subjectivity Sentence Classifiers from Unannotated Text

The labeled sentences identified by the rule-based classifiers provide us with the opportunity to apply supervised learning algorithms to our sentence classification task. Previous work [2], [30], [10] has found that naive Bayes [73] performs well for subjectivity recognition, and it is a simple and efficient algorithm to work with. Thus, we use naive Bayes as our learning algorithm.[9] A classifier developed using machine learning promises to have higher overall F-measure (albeit lower precision) than the rule-based classifiers described above (which are not able to classify as subjective or objective approximately 60 percent of the input corpus).

The purpose of developing a subjective/objective sentence classifier in this paper is to enable an IE system to consider sentence subjectivity when evaluating its extractions, as described in the following section. The IE task we address is the MUC-4 terrorism task. This data set is not manually annotated for subjectivity, and is likely to differ from the general news corpus that has been manually annotated for subjectivity. Thus, we use rule-based classifiers as described above in Section 3.4.1 to create MUC-4 training data, and then create a naive Bayes classifier by training on these data.

The naive Bayes classifier uses several types of set-valued features (some are based on features used in [72]). There is a feature for each of the following sets: the strongly subjective clues used by the original rule-based classifiers, including the nouns learned as described above in Section 3.3; the weakly subjective clues used by the original objective rule-based classifier (again, including the nouns learned as described above); the subjective patterns generated by the extraction pattern learner described in Section 3.4.2; and the objective patterns learned by the extraction pattern learner. We also add features for the following parts of speech, which have been shown to be effective in previous work [72], [10], [3]: pronouns, modals (excluding "will"), adjectives, cardinal numbers, and adverbs (excluding "not"). A three-valued feature is defined for each set based on the presence of 0, 1, or ≥ 2 members of that set in the sentence (we found such features to be effective in earlier work on supervised subjectivity classification [74], [30]). In addition, to incorporate contextual information in the classifier, another three-valued feature is defined for each set based on the presence of 0, 1, or ≥ 2 members of that set in the previous and next sentences combined.

The naive Bayes classifier uses a greater variety of features than the initial rule-based classifier, and it exploits a probabilistic model to make classification decisions based on combinations of its features. Thus, it can potentially label a larger and more diverse set of sentences in the unlabeled corpus more reliably than the rule-based classifiers can. In a self-training step, the system uses the naive Bayes classifier to relabel the training data that it started with, and then repeats the subsequent steps (extraction pattern learning and naive Bayes training).

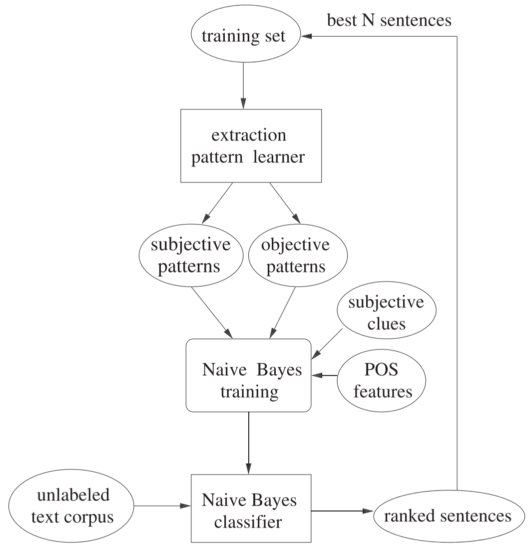

Fig. 6. The self-training process.

We adopt a conservative strategy, and use the naive Bayes classifier to label only the 90 percent of sentences it is most confident about. The remaining 10 percent are labeled as undecided and are ultimately treated as objective in the IE experiments described later.

The measure of confidence, CM, comes from the scores produced by the naive Bayes classifier (f_i is the ith feature used in the classifier):

$$CM = \Bigg| log(\Pr(subjective)) + \sum_i log(\Pr(f_i|subjective)) - (log(\Pr(objective)) + \sum_i log(\Pr(f_i|objective))) \Bigg|.$$

The overall process used to create the classifier used in the IE experiments described below is depicted in Fig. 6.

We would like to assess the performance of the classifier against manually annotated data. Since we do not have MUC data manually annotated for subjectivity, but only the MPQA news data, we create a classifier trained on data harvested from our unannotated store of news data (Section 3.4.1), and then evaluate it on the manually annotated MPQA test set used throughout this paper. The classifier achieves 72.7 percent subjective precision, 84.8 percent subjective recall, and 78.3 percent subjective F-measure. Thus, as we desired, we have a classifier with higher F-measure (a better balance of both precision and recall) than the rule-based classifiers originally used to create the training data.

In [30], we created a supervised classifier trained and tested on MPQA data. The F-measure of that system is 79.3. While this is not directly comparable to the F-measure given above, as the data are not exactly the same, it does show that the performance of the system here rivals the performance of the system trained on manually annotated

9. Future work could explore obtaining even better results with other learning algorithms such as SVM.

data. Note that other groups have created sentence-level classifiers trained on annotated data, but their data are too different to make reasonable comparisons.[10]

5 Improving Information Extraction with Subjectivity Classification

5.1 Motivation

The goal of information extraction (IE) systems is to extract facts related to events of interest from unstructured text. IE systems typically operate with tunnel vision, eager to extract any fact that appears to be relevant based on relatively local expressions and context. Consequently, IE systems often generate false hits from sentences that should not be taken literally. For example, IE systems can be easily misled by metaphorical language. Consider how the following sentences might be (mis)interpreted by an IE system that is searching for information about bombings and assaults:

1. *The Parliament exploded into fury against the government when word leaked out ...*
2. *D'Aubuisson unleashed harsh attacks on Duarte ...*

In sentence 1, the IE system might conclude that a bombing took place at the Parliament building. This is incorrect because the verb "exploded" is being used metaphorically. In sentence 2, the IE system might conclude that Duarte was the victim of a physical attack by D'Aubuisson, which is incorrect because "attacks" is describing a verbal tirade and not a physical assault.

Even news articles, which are generally assumed to be primarily factual in nature, frequently include opinions and evaluative statements by individuals, government officials, and organizations. Wiebe et al. [36], [37] studied a collection of *Wall Street Journal* news articles[11] and found that 44 percent of the sentences contain subjectivity. Sentence 3 below shows an example of an obvious opinion statement that could easily mislead an IE system into extracting "the economy" as a physical target. Opinion statements can also include unsupported allegations, rampant speculations, and hypothetical scenarios like the one in sentence 4, which may lead to an incorrect extraction reporting that a congressman was killed.

3. *The subversives must suspend the aggression against the people and the destruction of the economy ...*
4. *Searching congressmen is not very nice, but it would be worse if one were killed.*

Our hypothesis is that many incorrect extractions can be prevented by identifying sentences that contain subjective language and disallowing extractions from them. Sentences 3 and 4 above are examples of opinion statements that could be filtered before applying an IE system. Sentences 1 and 2 are also subjective in nature because sentence 1 describes negative emotions (*"exploded into fury"*) and sentence 2 describes a personal attack (*"harsh attacks"*).

In this section, we present a series of experiments to determine whether subjective sentence classification can improve the accuracy of information extraction systems. We explore several strategies for using subjectivity classifications, including an aggressive strategy that discards all extractions in subjective sentences and more complex strategies that selectively discard extractions.

5.2 The Information Extraction System

As the basis for our experiments, we constructed an IE system for the MUC-4 terrorism domain. The MUC-4 information extraction task [16] is to identify facts associated with terrorist events, such as bombings, kidnappings, and murders. The MUC-4 data set consists of 1,700 articles and associated answer key templates, which contain the correct information that should be extracted from each document. Roughly half of the documents mention a relevant terrorist event, while the other half do not (i.e., the answer key templates for these stories are empty).

We generated extraction patterns for this domain using the AutoSlog-TS extraction pattern learner [14]. AutoSlog-TS requires "relevant" and "irrelevant" texts for training, so we used the relevance judgments associated with the answer keys to partition the training set (see Section 5.3.1). We used the Sundance system [67] to parse the documents and apply the extraction patterns.

AutoSlog-TS is a weakly supervised learner that automatically generates a ranked list of extraction patterns and then a person must review the top-ranked patterns to decide which ones are useful for the IE task. The human reviewer manually assigns an event role to the patterns that are retained (e.g., the pattern "<subject> was killed" will extract victims, so would be assigned to the VICTIM template slot). AutoSlog-TS generated 40,553 distinct patterns from the MUC-4 training set. A person manually reviewed the top-ranked patterns that had a score ≥ 0.951[12] and frequency ≥ 3, and retained 397 for the MUC-4 IE task. We will refer to these 397 patterns as our *baseline* IE system.

5.3 Experimental Results

5.3.1 The Data Set

We conducted our experiments using the MUC-4 IE data set [16], which consists of 1,700 articles and their corresponding answer keys for the domain of Latin American terrorism. We evaluated performance on four of the MUC-4 *string* template slots, which require textual extractions: perpetrators (individuals), victims, physical targets, and weapons. The MUC-4 data set is divided into 1,300 development (DEV) texts, and four test sets of 100 texts each (TST1, TST2, TST3, and TST4).[13] We used 1,400 texts ($\mathrm{DEV} + \mathrm{TST1}$) as our training set, 100 texts (TST2) as a tuning set, and 200 texts ($\mathrm{TST3} + \mathrm{TST4}$) as our test set.

The IE process typically involves extracting information from sentences and then mapping that information into answer key templates, one template for each event described in the story. Many of the MUC-4 stories describe

10. Pang and Lee [43] automatically create a data set by treating all snippets from movie reviews as subjective and all sentences from plot summaries as objective. Yu and Hatzivassiloglou [10] and Dave et al. [3] manually label sentences with classes which cannot be directly mapped to our subjective/objective classification. The data in [75] are Japanese.

11. Editorial and review articles were explicitly removed.

12. This score corresponds to the RlogF metric, described in [14].

13. The DEV texts were used for development in MUC-3 and MUC-4. The TST1 and TST2 texts were used as test sets for MUC-3 and then as development texts for MUC-4. The TST3 and TST4 texts were used as the test sets for MUC-4.

TABLE 6
Subjectivity Filtering Results on the MUC-4 Test Set

System	Rec	Prec	F	Correct	Wrong
(a) IE	.52	.42	.47	266	367
(b) IE+Subj	.44	.44	.44	218 (-48)	273 (-94)
(c) IE+Subj_Attr	.46	.44	.45	231 (-35)	289 (-78)
(d) IE+Subj_Attr_Slct	.51	.45	.48	258 (-8)	311 (-56)
(e) IE+Subj_Attr_Slct+SubjEP	.51	.46	.48	258 (-8)	305 (-62)

multiple terrorist events, and therefore should generate multiple output templates. Template generation is a complex process, requiring coreference resolution and discourse analysis to determine how many incidents were reported and which facts belong with each incident. Our goal is to use subjectivity filtering to eliminate bad extractions immediately so that the discourse analyzer doesn't have to grapple with them. Consequently, we evaluated the performance of our IE system on the extractions themselves, before template generation would take place. This approach directly measures how well we are able to filter bad extractions without introducing confounding factors from the template generation process.[14]

5.3.2 Baseline Results

Row (a) of Table 6 shows the results of our baseline IE system on the MUC-4 test set. The first three columns show Recall, Precision, and F-measure ($\beta = 1$) scores. The last two columns show the number of correct extractions and the number of incorrect extractions. The IE system achieved 52 percent recall with 42 percent precision, yielding an F-measure of 47 percent.[15] These scores are similar to those of other IE systems on the MUC-4 IE task [13], [16], but they are not directly comparable because those evaluations include template generation. For the purposes of the current study, these numbers simply represent a baseline against which we will compare different strategies for reducing the number of bad extractions by filtering subjective sentences.

5.3.3 Aggressive Subjective Sentence Filtering

For our first attempt at subjectivity filtering, we discarded *all* extractions that occurred in sentences labeled as subjective by the classifier. Row (b) in Table 6 shows these results. Precision increased $+2$ percent because 94 bad extractions were discarded, but recall dropped -8 percent because 48 correct extractions were also discarded. These results seemed to confirm our intuition that many bad extractions occur in subjective sentences, but we were surprised to find that so many good extractions also come from subjective sentences. Clearly, the strategy of indiscriminately discarding all extractions in subjective sentences is too simplistic, so we decided to further investigate how subjective statements and factual statements coexist in the same sentence.

14. For example, suppose that a coreference resolver incorrectly decides that two extractions are coreferent and merges them. One extraction will be lost, but the fault lies with the coreference resolver and not with the extraction engine per se.

15. We used a *head noun* scoring scheme, where we scored an extraction as correct if its head noun matched the head noun in the answer key. This approach allows for different leading modifiers in an NP as long as the head noun is the same. For example, "armed men" will successfully match "five armed men." We also discarded pronouns (they weren't scored at all) because our system does not perform coreference resolution.

5.3.4 Source Attribution Modification

An immediate observation was that many sentences with source attributions contain factual information, and these were often being classified as subjective. News articles, in particular, often report information by citing a source (e.g., *"The Associated Press reported . . . "* or *"The President stated . . . "*). Many of the same verbs, however, can also be used to attribute an opinionated statement to someone or something (e.g., *CNN reported that Bush praised his Attorney General for . . .* or *Bush stated that he was opposed to . . .*). Nevertheless, the presence of a source attribution in a sentence is a strong indicator that the sentence may contain important factual information, so we decided to override the subjectivity classifier when a source attribution occurs in a sentence that had a modest subjectivity score. More precisely, we modified our system to override the classifier and extract information from a sentence when it satisfies the following two criteria: 1) The confidence measure, CM, is ≤ 25, indicating that the classifier considers the sentence to be only weakly subjective, and 2) the sentence contains any of the following attribution verbs: {*affirm, announce, cite, confirm, convey, disclose, report, tell, say, state*}. Row (c) of Table 6 shows the results of the modified system (IE+Subj_Attr). Extracting information from source attribution sentences improved recall $+2$ percent, while maintaining the same level of precision.

5.3.5 Selective Subjective Sentence Filtering

Further investigation revealed that facts and opinions frequently do coexist in the same sentence, so we simply can't ignore all sentences that contain subjective language. For example, consider the sentence: *"He was outraged by the terrorist attack on the World Trade Center."* "Outraged" is a highly subjective term. Nonetheless, this sentence also mentions a pertinent fact: There was a terrorist attack on the World Trade Center. Our solution was to modify our system to recognize "mixed" sentences that are likely to contain both facts and opinions. Our approach was to identify *indicator* patterns that should always be allowed to extract information, regardless of whether they appear in a subjective context or not. Intuitively, an *indicator* pattern represents an expression that is virtually a dead giveaway that a fact of interest is present. While no patterns are perfectly reliable, indicator patterns tend to be much more reliable than other patterns. For example, *"murder of <NP>"* and *"<NP> was assassinated"* nearly always identify murder victims, regardless of the surrounding context. In contrast, *nonindicator* patterns represent expressions that may or may not extract relevant information depending on the context. For example, *"<NP> was arrested"* and *"injured by <NP>"* may extract the names of terrorists when these patterns appear in a terrorist event description, but they may extract other information when they appear in other contexts.

To automatically recognize indicator patterns, we used the statistics generated by AutoSlog-TS during training. If a pattern has a conditional probability $P(relevant \mid pattern_i) \geq .65$ and a frequency ≥ 10, then we label it as an *indicator* pattern because it is highly correlated with domain-relevant texts. Otherwise, we label the pattern as a *nonindicator pattern*. We conducted an initial experiment to see if the indicator patterns alone would be sufficient for our IE task. When using only the indicator patterns in our baseline IE system, recall dropped from 52 to 40 percent, which shows that the nonindicator patterns do extract a lot of relevant information as well.

We then modified our system to perform *selective subjectivity filtering*: Extractions from indicator patterns are never discarded, but extractions from nonindicator patterns are discarded if they appear in a subjective sentence. Row (d) of Table 6 shows the results. Selective subjectivity filtering had a dramatic impact on performance, recovering 27 correct extractions that were previously discarded, which brought recall up from 46 to 51 percent (close to the 52 percent maximum recall possible without any subjectivity filtering) while further improving precision (to 45 percent).

5.3.6 Subjective Extraction Pattern Filtering

The terrorism domain patterns in our IE system were manually reviewed (see Section 5.2), so they should be of high quality. But anticipating which patterns will perform well is difficult because it is hard for people to anticipate the kinds of expressions that they will match and the contexts in which they will appear. We hypothesized that subjectivity analysis may be helpful in providing an empirical, alternative assessment of each pattern, not just in terms of relevance to the domain, but in terms of whether it is more frequently used in subjective or objective contexts. Intuitively, if a pattern is highly correlated with subjective sentences, then perhaps we should be suspicious that the pattern is matching different kinds of expressions or is being used in a different way than was expected.

To investigate this idea, we applied the subjective sentence classifier and the nonindicator extraction patterns to the training set and counted the number of times each pattern occurred in subjective versus objective sentences. For each pattern, we estimated the probability that a sentence is subjective given that it contains that pattern. We deemed an extraction pattern to be a *subjective pattern* if $P(subjective \mid pattern_i) > .50$ and its frequency ≥ 10.[16] Ten of our IE patterns were identified as being subjective patterns:

attacks on <np>	to attack <dobj>
communique by <np>	to destroy <dobj>
<subj> was linked	leaders of <np>
<subj> unleashed	was aimed at <np>
offensive against <np>	dialogue with <np>

The pattern *"was aimed at <np>"* illustrates how an expression can be used in multiple ways and why it can be difficult to predict which usage will be more common. A person might expect this pattern to reliably extract targets (e.g., *"One attack was aimed at fuel storage tanks."*), but the statistics revealed that 58 percent of the time this expression occurred in subjective contexts with a more general use of the expression (e.g., *"The proposal is aimed at circumventing the skepticism of the Board."*).

Next, we modified our system to ignore all extractions from these "subjective patterns" (even though they were previously deemed to be reliable IE patterns by a human). Row (e) of Table 6 shows the results. This process filtered six additional extractions, all of which were incorrect. Although the precision improvement was small, using automated subjectivity analysis to reassess manually reviewed patterns seems to be valuable: It costs nothing and adds an additional source of quality control to the IE process.

Our final system achieved a precision gain of $+4$ percent over the baseline IE system, with minimal recall loss (-1 percent). In absolute terms, using the subjectivity filtering strategies resulted in 62 fewer incorrect extractions while losing only eight correct extractions. Table 7 breaks down the individual results for the four types of extracted information. Subjectivity filtering improved performance in all four categories, increasing precision by as much as $+5$ percent on two of the four categories.

TABLE 7
IE Results for Individual Slots

	IE		IE w/Subj	
Category	**Rec**	**Prec**	**Rec**	**Prec**
Perpetrator	.47	.33	.45	.38
Victim	.51	.50	.50	.52
Target	.63	.42	.62	.47
Weapon	.45	.39	.43	.42
Total	.52	.42	.51	.46

5.4 Combining Subjectivity Classification with Topic Classification

As we explained in Section 5.3.1, the MUC-4 corpus is a mixture of relevant (on-topic) texts and irrelevant (off-topic) texts that do not mention any terrorist events. We wondered whether subjectivity filtering was eliminating bad extractions primarily from the irrelevant texts. If so, then a good topic-based classifier might suffice and eliminate the need for subjectivity filtering.

To answer this question, we conducted an experiment to see how subjectivity filtering would perform if we had a perfect topic-based text classifier. The first row of Table 8 shows the results of applying our baseline IE system only to the relevant texts in the test set. Precision increases by $+11$ percent compared to the results on the entire test set. This shows that many bad extractions were indeed eliminated by removing the off-topic texts. However, the second row of Table 8 shows the results of applying our IE system with subjectivity filtering only to the relevant texts. Precision improves by $+3$ percent over the baseline system, which is almost the same level of precision improvement that we saw on the complete test set. These results demonstrate that subjectivity filtering is in fact eliminating bad extractions from relevant (on-topic) documents as well. Our conclusion is that topic-based text filtering and subjectivity filtering are complementary: Topic-based filtering will improve precision, but subjectivity filtering combined with topic-based filtering performs even better.

16. In our corpus, we observed that the subjectivity classifier labeled about 50 percent of the sentences as subjective. So, we made the assumption that there is roughly a 50/50 split between subjective and objective sentences.

TABLE 8
IE Results on the Relevant Texts Only

System	Rec	Prec
IE	.52	.53
IE w/Subj	.51	.56

6 Conclusions

The demand both for high-quality information extraction and high-quality subjectivity analysis systems continues to grow as increasingly vast amounts of text are available online and as social interaction increasingly occurs online. In this paper, we explore a perhaps surprising synergy between subjectivity analysis and information extraction. First, we investigate exploiting IE techniques to learn and recognize subjective language. Though several subjectivity lexicons exist, they are far from complete; in addition, subjective language may be domain dependent. A goal of our work is to create subjective language learners that do not require manually annotated data as input, which may be applied to large amounts of text to create richer dictionaries and which may be applied to domains for which manually annotated data do not exist. Our approach is to use weakly supervised IE learning methods to automatically discover subjective words and expressions from unannotated texts. In learning subjective nouns using extraction pattern contexts, we found that both the Meta-Bootstrapping and Basilisk algorithms are effective and that both find words that the other does not. After a manual review of 3-4 hours, over 1,000 subjective nouns were learned. Evaluated against a gold standard, the probability that a sentence which has one is subjective is high (up to 0.87 for those judged strongly subjective).

We added the learned nouns to an existing subjectivity lexicon, and then use rule-based classifiers to bootstrap from the lexicon to harvest subjective and objective sentences from unannotated data. The precision of the classifiers is high (over 0.90 for the subjective classifier and over 0.80 for the objective one). The harvested data are fed as input into an extraction pattern learning algorithm. Patterns that are variations of each other may distribute differently in subjective versus objective texts, and it is perhaps impossible for a human to predict such differences. Thus, these patterns are not reviewed by a human; we propose that they be used as evidence during subjectivity classification rather than added as entries to a subjectivity lexicon. Even without human review, the probability that a sentence which has one is subjective is over 0.80. For both the learned nouns and the learned extraction patterns, we provide evidence that they represent additional knowledge over and above the knowledge that was already contained in the subjectivity lexicon.

The harvested training data are used to train a subjective/objective sentence classifier which combines features from previous work with features built from the nouns and patterns learned in this paper. The purpose of the classifier in this paper is to enable an IE system to consider sentence subjectivity on data that are not manually annotated for subjectivity. The classifier achieves an F-measure over 0.78, which rivals classifiers trained on manually annotated data.

Overall, the required inputs for our work on subjectivity are an existing subjectivity lexicon, a set of seed nouns, and a small amount of human review.

The classifier is then used in our second research direction: the use of subjectivity analysis to improve the accuracy of fact-based information extraction systems. Information extraction systems suffer from false hits, and we observed that many of these false hits occur in subjective sentences. Our hypothesis was that many incorrect extractions can be prevented by identifying sentences that contain subjective language and disallowing extractions from them. We explore the idea of using a subjective sentence classifier, on data which have not been annotated for subjectivity, to proactively identify and filter subjective sentences before extracting information from them. We carried out a series of experiments exploring several strategies, including an aggressive strategy that discards all extractions in subjective sentences, and more complex ones that are more selective. The simple aggressive strategy does increase precision, but at the cost of too much loss in recall. With a combination of more complex filtering strategies, the performance of the IE system is improved by realizing increases in precision with little decrease in recall. We also provide evidence that topic-based text filtering and subjectivity filtering are complementary ways to increase information extraction performance.

Acknowledgments

This research was supported in part by the US National Science Foundation (NSF) under grants IIS-0208798 and IIS-0208985, and this material is based upon research supported by the Advanced Research and Development Activity (ARDA) under Contract No. NBCHC040012. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of ARDA or the Department of Interior-National Business Center (DOI-NBC). The authors are very grateful to Theresa Wilson for her invaluable programming support and contributions to data preparation, Bill Phillips for his work on the IE component, and to the anonymous reviewers for their helpful comments and suggestions.

References

[1] P. Turney and M.L. Littman, "Measuring Praise and Criticism: Inference of Semantic Orientation from Association," *ACM Trans. Information Systems*, vol. 21, no. 4, pp. 315-346, 2003.

[2] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs Up? Sentiment Classification Using Machine Learning Techniques," *Proc. Conf. Empirical Methods in Natural Language Processing*, pp. 79-86, 2002.

[3] K. Dave, S. Lawrence, and D.M. Pennock, "Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews," *Proc. 12th Int'l World Wide Web Conf.*, http://www2003.org, 2003.

[4] T. Nasukawa and J. Yi, "Sentiment Analysis: Capturing Favorability Using Natural Language Processing," *Proc. Second Int'l Conf. Knowledge Capture*, pp. 70-77, 2003.

[5] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima, "Mining Product Reputations on the Web," *Proc. Eighth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pp. 341-349, 2002.

[6] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack, "Sentiment Analyzer: Extracting Sentiments about a Given Topic Using Natural Language Processing Techniques," *Proc. Third IEEE Int'l Conf. Data Mining*, pp. 427-434, 2003.

[7] E. Spertus, "Smokey: Automatic Recognition of Hostile Messages," *Proc. Eighth Ann. Conf. Innovative Applications of Artificial Intelligence,* pp. 1058-1065, 1997.

[8] S.R. Das and M.Y. Chen, "Yahoo! for Amazon: Opinion Extraction from Small Talk on the Web," *Proc. Eighth Asia Pacific Finance Assoc. Ann. Conf.*, 2001.

[9] R. Tong, "An Operational System for Detecting and Tracking Opinions in On-Line Discussions," *Proc. SIGIR Workshop Operational Text Classification,* pp. 1-6, 2001.

[10] H. Yu and V. Hatzivassiloglou, "Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences," *Proc. Conf. Empirical Methods in Natural Language Processing,* pp. 129-136, 2003.

[11] S. Patwardhan and E. Riloff, "A Unified Model of Phrasal and Sentential Evidence for Information Extraction," *Proc. Conf. Empirical Methods in Natural Language Processing,* 2009.

[12] S. Patwardhan and E. Riloff, "Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions," *Proc. Conf. Empirical Methods in Natural Language Processing,* 2007.

[13] H. Chieu, H. Ng, and Y. Lee, "Closing the Gap: Learning-Based Information Extraction Rivaling Knowledge-Engineering Methods," *Proc. Ann. Meeting Assoc. for Computational Linguistics,* 2003.

[14] E. Riloff, "Automatically Generating Extraction Patterns from Untagged Text," *Proc. 13th Nat'l Conf. Artificial Intelligence,* pp. 1044-1049, 1996.

[15] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert, "CRYSTAL: Inducing a Conceptual Dictionary," *Proc. 14th Int'l Joint Conf. Artificial Intelligence,* pp. 1314-1319, 1995.

[16] MUC-4 Proc., *Proc. Fourth Message Understanding Conf.*, 1992.

[17] R. Grishman, S. Huttunen, and R. Yangarber, "Real-Time Event Extraction for Infectious Disease Outbreaks," *Proc. Human Language Technology Conf.*, 2002.

[18] MUC-7 Proc., *Proc. Seventh Message Understanding Conf.*, 1998.

[19] R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen, "Automatic Acquisition of Domain Knowledge for Information Extraction," *Proc. 18th Int'l Conf. Computational Linguistics,* 2000.

[20] MUC-5 Proc., *Proc. Fifth Message Understanding Conf.*, 1993.

[21] D. Freitag, "Toward General-Purpose Learning for Information Extraction," *Proc. 36th Ann. Meeting Assoc. for Computational Linguistics,* 1998.

[22] D. Freitag and A. McCallum, "Information Extraction with HMM Structures Learned by Stochastic Optimization," *Proc. 17th Nat'l Conf. Artificial Intelligence,* pp. 584-589, Aug. 2000.

[23] M.E. Califf and R.J. Mooney, "Relational Learning of Pattern-Match Rules for Information Extraction," *Proc. ACL Workshop Natural Language Learning,* pp. 9-15, 1997.

[24] F. Ciravegna, "Adaptive Information Extraction from Text by Rule Induction and Generalisation," *Proc. 17th Int'l Joint Conf. Artificial Intelligence,* 2001.

[25] V. Hatzivassiloglou and K. McKeown, "Predicting the Semantic Orientation of Adjectives," *Proc. 35th Ann. Meeting Assoc. for Computational Linguistics,* pp. 174-181, 1997.

[26] J. Kamps and M. Marx, "Words with Attitude," *Proc. First Int'l WordNet Conf.*, pp. 332-341, 2002.

[27] T. General-Inquirer, http://www.wjh.harvard.edu/~inquirer/spreadsheet_guide.htm, 2000.

[28] J. Wiebe and E. Riloff, "Creating Subjective and Objective Sentence Classifiers from Unannotated Texts," *Proc. Sixth Int'l Conf. Intelligent Text Processing and Computational Linguistics,* pp. 486-497, 2005.

[29] K. Bloom, N. Garg, and S. Argamon, "Extracting Appraisal Expressions," *Proc. Human Language Technologies: Conf. North Am. Chapter of the Assoc. for Computational Linguistics,* pp. 308-315, 2007.

[30] E. Riloff, J. Wiebe, and T. Wilson, "Learning Subjective Nouns Using Extraction Pattern Bootstrapping," *Proc. Seventh Conf. Natural Language Learning,* pp. 25-32, 2003.

[31] E. Riloff and J. Wiebe, "Learning Extraction Patterns for Subjective Expressions," *Proc. Conf. Empirical Methods in Natural Language Processing,* pp. 105-112, 2003.

[32] E. Riloff, J. Wiebe, and W. Phillips, "Exploiting Subjectivity Classification to Improve Information Extraction," *Proc. 20th Nat'l Conf. Artificial Intelligence,* pp. 1106-1111, 2005.

[33] J. Wiebe, T. Wilson, and C. Cardie, "Annotating Expressions of Opinions and Emotions in Language," *Language Resources and Evaluation,* vol. 39, nos. 2/3, pp. 165-210, 2005.

[34] M. Thelen and E. Riloff, "A Bootstrapping Method for Learning Semantic Lexicons Using Extraction Pattern Contexts," *Proc. Conf. Empirical Methods in Natural Language Processing,* 2002.

[35] E. Riloff and R. Jones, "Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping," *Proc. 16th Nat'l Conf. Artificial Intelligence,* pp. 474-479, 1999.

[36] J. Wiebe, T. Wilson, and M. Bell, "Identifying Collocations for Recognizing Opinions," *Proc. ACL-01 Workshop Collocation: Computational Extraction, Analysis and Exploitation,* pp. 24-31, 2001.

[37] J. Wiebe, T. Wilson, R. Bruce, M. Bell, and M. Martin, "Learning Subjective Language," *Computational Linguistics,* vol. 30, no. 3, pp. 277-308, 2004.

[38] J. Wiebe and R. Mihalcea, "Word Sense and Subjectivity," *Proc. Int'l Conf. Computational Linguistics and 44th Ann. Meeting Assoc. for Computational Linguistics,* 2006.

[39] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis," *Proc. Human Language Technologies Conf./Conf. Empirical Methods in Natural Language Processing,* pp. 347-354, 2005.

[40] H. Takamura, T. Inui, and M. Okumura, "Latent Variable Models for Semantic Orientations of Phrases," *Proc. 11th Meeting of the European Chapter of the Assoc. for Computational Linguistics,* 2006.

[41] A. Andreevskaia and S. Bergler, "Sentiment Tag Extraction from Wordnet Glosses," *Proc. Fifth Int'l Conf. Language Resources and Evaluation,* 2006.

[42] A. Esuli and F. Sebastiani, "Determining Term Subjectivity and Term Orientation for Opinion Mining," *Proc. 11th Meeting of the European Chapter Assoc. for Computational Linguistics,* http://tcc.itc.it/projects/ontotext/Publications/EACL-2006-Esuli-Sebastiani.pdf, 2006.

[43] B. Pang and L. Lee, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts," *Proc. 42nd Ann. Meeting Assoc. for Computational Linguistics,* pp. 271-278, 2004.

[44] S.-M. Kim and E. Hovy, "Identifying and Analyzing Judgment Opinions," *Proc. Conf. Human Language Technology Conf. North Am. Assoc. for Computational Linguistics,* pp. 200-207, http://www.aclweb.org/anthology/N/N06/N06-1026, 2006.

[45] P. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews," *Proc. 40th Ann. Meeting Assoc. for Computational Linguistics,* pp. 417-424, 2002.

[46] M. Gamon and A. Aue, "Automatic Identification of Sentiment Vocabulary: Exploiting Low Association with Known Sentiment Terms," *Proc. ACL-05 Workshop Feature Eng. for Machine Learning in Natural Language Processing,* 2005.

[47] H. Kanayama and T. Nasukawa, "Fully Automatic Lexicon Expansion for Domain-Oriented Sentiment Analysis," *Proc. Conf. Empirical Methods in Natural Language Processing,* pp. 355-363, 2006.

[48] J. Wiebe and T. Wilson, "Learning to Disambiguate Potentially Subjective Expressions," *Proc. Sixth Conf. Natural Language Learning,* pp. 112-118, 2002.

[49] H. Takamura, T. Inui, and M. Okumura, "Extracting Semantic Orientations of Phrases from Dictionary," *Proc. Human Language Technologies: Conf. North Am. Chapter Assoc. for Computational Linguistics,* pp. 292-299, http://www.aclweb.org/anthology/N/N07/N07-1037, Apr. 2007.

[50] Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan, "Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns," *Proc. Human Language Technologies Conf./Conf. Empirical Methods in Natural Language Processing,* pp. 355-362, 2005.

[51] S.-M. Kim and E. Hovy, "Extracting Opinions, Opinion Holders and Topics Expressed in Online News Media Text," *Proc. ACL Workshop Sentiment and Subjectivity in Text,* pp. 1-8, http://www.aclweb.org/anthology/W/W06/W06-0301, 2006.

[52] H. Takamura, T. Inui, and M. Okumura, "Extracting Semantic Orientations of Words Using Spin Model," *Proc. 43rd Ann. Meeting Assoc. for Computational Linguistics,* http://acl.ldc.upenn.edu/P/P05/P05-1017.pdf, 2005.

[53] T. Zagibalov and J. Carroll, "Automatic Seed Word Selection for Unsupervised Sentiment Classification of Chinese Text," *Proc. 22nd Int'l Conf. Computational Linguistics,* pp. 1073-1080, http://www.aclweb.org/anthology/C08-1135, Aug. 2008.

[54] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, Bollywood, Boom-Boxes and Blenders: Domain Adaptation for Sentiment Classification," *Proc. 45th Ann. Meeting Assoc. for Computational Linguistics,* pp. 440-447, http://www.aclweb.org/anthology/P/P07/P07-1056, June 2007.

[55] A. Andreevskaia and S. Bergler, "When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging," *Proc. Ann. Meeting Assoc. for Computational Linguistics,* pp. 290-298, http://www.aclweb.org/anthology/P/P08/P08-1034, 2008.
[56] W. Phillips and E. Riloff, "Exploiting Role-Identifying Nouns and Expressions for Information Extraction," *Proc. Int'l Conf. Recent Advances in Natural Language Processing,* 2007.
[57] R. Bunescu and R. Mooney, "Collective Information Extraction with Relational Markov Networks," *Proc. 42nd Ann. Meeting Assoc. for Computational Linguistics,* pp. 438-445, July 2004.
[58] J. Kim and D. Moldovan, "Acquisition of Semantic Patterns for Information Extraction from Corpora," *Proc. Ninth IEEE Conf. Artificial Intelligence for Applications,* pp. 171-176, 1993.
[59] S. Huffman, "Learning Information Extraction Patterns from Examples," *Proc. Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing,* S. Wermter, E. Riloff, and G. Scheler, eds., pp. 246-260, 1996.
[60] E. Riloff, "Automatically Constructing a Dictionary for Information Extraction Tasks," *Proc. 11th Nat'l Conf. Artificial Intelligence,* pp. 811-816, 1993.
[61] M.E. Califf, "Relational Learning Techniques for Natural Language Information Extraction," Technical Report AI98-276, Artificial Intelligence Laboratory, The Univ. of Texas at Austin, 1998.
[62] E. Agichtein and L. Gravano, "Snowball: Extracting Relations from Large Plain-Text Collections," *Proc. Fifth ACM Conf. Digital Libraries,* pp. 85-94, June 2000.
[63] K. Sudo, S. Sekine, and R. Grishman, "An Improved Extraction Patterns Representation Model for Automatic IE Pattern Acquisition," *Proc. 41st Ann. Meeting Assoc. for Computational Linguistics,* pp. 224-231, July 2003.
[64] A. Yakushiji, Y. Miyao, T. Ohta, Y. Tateisi, and J. Tsujii, "Construction of Predicate-Argument Structure Patterns for Biomedical Information Extraction," *Proc. Conf. Empirical Methods in Natural Language Processing,* pp. 284-292, July 2006.
[65] A. Popescu, A. Yates, and O. Etzioni, "Class Extraction from the World Wide Web," *Proc. AAAI Workshop Adaptive Text Extraction and Mining,* I. Muslea, ed., pp. 68-73, July 2004.
[66] B. Klebanov, K. Knight, and D. Marcu, "Text Simplification for Information Seeking Applications," *On the Move to Meaningful Internet Systems,* pp. 735-747, Springer-Verlag, 2004.
[67] E. Riloff and W. Phillips, "An Introduction to the Sundance and AutoSlog Systems," Technical Report UUCS-04-015, School of Computing, Univ. of Utah, 2004.
[68] B. Levin, *English Verb Classes and Alternations: A Preliminary Investigation.* Univ. of Chicago Press, 1993.
[69] T. Ballmer and W. Brennenstuhl, *Speech Act Classification: A Study in the Lexical Analysis of English Speech Activity Verbs.* Springer-Verlag, 1981.
[70] Framenet, http://www.icsi.berkeley.edu/~framenet/, 2002.
[71] J. Wiebe, "Tracking Point of View in Narrative," *Computational Linguistics,* vol. 20, no. 2, pp. 233-287, 1994.
[72] J. Wiebe, R. Bruce, and T. O'Hara, "Development and Use of a Gold Standard Data Set for Subjectivity Classifications," *Proc. 37th Ann. Meeting Assoc. for Computational Linguistics,* pp. 246-253, 1999.
[73] T. Mitchell, *Machine Learning.* McGraw-Hill, 1997.
[74] J. Wiebe, R. Bruce, and T. O'Hara, "Development and Use of a Gold Standard Data Set for Subjectivity Classifications," *Proc. 37th Ann. Meeting Assoc. for Computational Linguistics,* pp. 246-253, 1999.
[75] N. Hiroshima, S. Yamada, O. Furuse, and R. Kataoka, "Searching for Sentences Expressing Opinions by Using Declaratively Subjective Clues," *Proc. ACL Workshop Sentiment and Subjectivity in Text,* pp. 39-46, http://www.aclweb.org/anthology/W/W06/W06-0306, 2006.

Janyce Wiebe is a professor of computer science and the Co-Director of the Intelligent Systems Program at the University of Pittsburgh. Her research with students and colleagues has been in discourse processing, pragmatics, and word-sense disambiguation. A main concentration of her research is "subjectivity analysis," recognizing and interpreting expressions of opinions and sentiments in text to support NLP applications such as question answering, information extraction, text categorization, and summarization. Her current and past professional roles include ACL Program Co-Chair, NAACL Program Chair, NAACL Executive Board member, *Computational Linguistics* and *Language Resources and Evaluation* Editorial Board member, AAAI Workshop Co-Chair, ACM Special Interest Group on Artificial Intelligence (SIGART) Vice-Chair, and ACM-SIGART/AAAI Doctoral Consortium Chair.

Ellen Riloff is an associate professor of computer science in the School of Computing at the University of Utah. Her primary research areas are information extraction, semantics, coreference resolution, and subjectivity analysis. A major emphasis of her research has been automatically acquiring the knowledge needed for natural language processing using bootstrapping methods that can learn from unannotated texts with minimal human supervision. Her professional roles have included service on the NAACL Executive Board, Human Language Technology (HLT) Advisory Board, DARPA/NSF Question Answering Roadmap Committee, *Computational Linguistics* Editorial Voard member, CoNLL Program Co-Chair, and as a faculty advisor for the ACL Student Research Workshop.