In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05)*

Exploiting Subjectivity Classification to Improve Information Extraction

Ellen Riloff
School of Computing
University of Utah
Salt Lake City, UT 84112
riloff@cs.utah.edu

Janyce Wiebe
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260
wiebe@cs.pitt.edu

William Phillips
School of Computing
University of Utah
Salt Lake City, UT 84112
phillips@cs.utah.edu

Abstract

Information extraction (IE) systems are prone to false hits for a variety of reasons and we observed that many of these false hits occur in sentences that contain subjective language (e.g., opinions, emotions, and sentiments). Motivated by these observations, we explore the idea of using subjectivity analysis to improve the precision of information extraction systems. In this paper, we describe an IE system that uses a subjective sentence classifier to filter its extractions. We experimented with several different strategies for using the subjectivity classifications, including an aggressive strategy that discards all extractions found in subjective sentences and more complex strategies that selectively discard extractions. We evaluated the performance of these different approaches on the MUC-4 terrorism data set. We found that indiscriminately filtering extractions from subjective sentences was overly aggressive, but more selective filtering strategies improved IE precision with minimal recall loss.

Introduction

The goal of information extraction (IE) systems is to extract facts related to a particular domain from natural language texts. IE systems typically operate with tunnel vision, eager to extract any fact that appears to be relevant based on relatively simple lexico-syntactic patterns. These patterns represent localized expressions so they are prone to false hits in sentences that should not be taken literally. For example, IE systems can be easily misled by colorful language that contains metaphor or hyperbole. Imagine what would happen if an IE system looking for information about bombings and physical assaults were applied to the sentences below:

(a) *The Parliament exploded into fury against the government when word leaked out ...*
(b) *D'Aubuisson unleashed harsh attacks on Duarte ...*

In sentence (a), the IE system may report that a bombing took place and "The Parliament" was the target of the bombing. This is incorrect because the verb "exploded" is being used metaphorically. In sentence (b), the IE system would probably report that Duarte was the victim of a physical attack by D'Aubuisson. This is also incorrect because "attacks" is metaphorically referring to a verbal tirade, not a physical assault.

Many documents also contain more than just factual information. News articles, for example, often include opinions and personal statements by individuals, government officials, and organizations. For example, sentence (c) is clearly an opinion statement that could 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 (d), which may lead to an incorrect extraction saying that a congressman was killed.

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

Our observation is that many incorrect extractions could be prevented by identifying sentences that contain subjective language and filtering extractions from them. *Subjective* sentences are sentences that express or describe opinions, evaluations, or emotions (Wiebe *et al.* 2004). For example, sentences (c) and (d) are obvious opinion statements. Sentences (a) and (b) are also subjective, (a) because it describes negative emotions and (b) because it describes a personal attack. Note that (a) and (b) are also metaphorical. Subjective sentences frequently contain metaphors and hyperbole.

In this paper, we explore the idea of using subjectivity analysis to improve the precision of information extraction systems by automatically filtering extractions that appear in subjective sentences. We experimented with different strategies for using the subjectivity classifications, including an aggressive strategy that discards all extractions in subjective sentences and more complex strategies that selectively discard extractions.

This paper is organized as follows. First, we review related work. Next, we describe the classifier that we use to automatically classify sentences as subjective or objective, and the IE system used in our experiments. Finally, we describe experiments involving several filtering strategies.

Related Work

There has been a recent swell of interest in automatically identifying opinions, emotions, evaluations, and sentiments in text. Such processing has been applied to many applications, including classification of reviews as positive or negative (e.g., (Turney & Littman 2003; Dave, Lawrence, & Pennock 2003; Pang & Lee 2004), recognizing hostile messages (e.g., (Spertus 1997)), analyzing product reputations (e.g., (Morinaga *et al.* 2002; Yi *et al.* 2003)), tracking sentiments toward events (e.g., (Tong 2001)), genre classification (e.g., (Yu & Hatzivassiloglou 2003; Wiebe *et al.* 2004)), mining and summarizing reviews (Hu & Liu 2004), multi-document summarization and question answering (e.g., (Yu & Hatzivassiloglou 2003)).

Information extraction systems have been developed for a variety of domains, including terrorism (MUC-4 Proceedings 1992; Chieu, Ng, & Lee 2003; Riloff 1996; Soderland *et al.* 1995), management succession (Yangarber *et al.* 2000), corporate acquisitions (Freitag 1998), job postings (Califf & Mooney 1997; Freitag & McCallum 2000), rental ads (Soderland 1999; Ciravegna 2001), seminar announcements (Ciravegna 2001; Freitag & McCallum 2000), and disease outbreaks (Grishman, Huttunen, & Yangarber 2002). This paper presents the first research effort to exploit subjectivity analysis to improve the performance of an information extraction system.

Klebanov et al. (2004) present 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.

The Subjectivity Classifier

Many systems that perform subjectivity analysis or related tasks work at the document level, for example classifying entire reviews as positive or negative (Turney & Littman 2003; Pang & Lee 2004). In contrast, we use a system developed by (Wiebe & Riloff 2005) that performs sentence-level subjectivity classification. Sentence-level classification is useful because most documents contain a mix of subjective and objective sentences. For example, newspaper articles are typically thought to be relatively objective, but Wiebe et al. (2004) reported that 44% of sentences in their corpus (in articles that are not editorials or reviews) are subjective.

Almost all systems that perform sentence-level classification require labeled training data as input (Yu & Hatzivassiloglou 2003; Dave, Lawrence, & Pennock 2003; Pang & Lee 2004). One of the main advantages of the system that we use is that it does not require labeled training data. Even without labeled data, experiments have shown that its performance rivals that of the best supervised learning systems (Wiebe & Riloff 2005).[1] This system applies a rule-based classifier to an unlabeled corpus to create training data, which is then used to train a Naive Bayes classifier.

First, the rule-based classifier is applied to an unlabeled corpus. This classifier consults a large list of well-established general subjectivity clues that have been published in the literature. If a sentence contains at least two clues, then the sentence is labeled as subjective. If a sentence contains none of the clues, and the sentence and surrounding context have sufficiently few subjectivity indicators, then the sentence is labeled as objective. Otherwise, the sentence is not given a label. This rule-based classifier achieves high precision but low recall.[2]

The sentences that are labeled by the rule-based classifier are then used as training data for a subsequent learning phase. The training data is first used to automatically learn extraction patterns that are associated with subjectivity. The AutoSlog-TS extraction pattern learner (Riloff 1996), described later, is used. (However, the subjectivity classifier's use of extraction patterns is completely unrelated to the IE application task that is the main focus of this paper.) Extraction patterns are used to represent subjective expressions because they are linguistically richer and more flexible than single words or N-grams. For example, *<subj> dealt blow* is an extraction pattern that matches all active voice verb phrases that have head = *dealt* and a direct object with head = *blow*, such as the main verb phrase of sentence (e):

> (e) *This act of hubris was dealt a revealingly swift blow by the completion of the Empire State Building a few months later.*

The classifier does not perform information extraction, but it simply uses the extraction patterns to match subjective expressions, ignoring the extracted noun phrases.

A Naive Bayes classifier (Mitchell 1997) is then constructed using the training data labeled by the rule-based classifier. The features of the Naive Bayes classifier are defined as counts of (1) the clues used by the rule-based classifier, (2) the expressions matched by the extraction patterns learned in the previous step, and (3) pronouns, modals, adjectives, cardinal numbers, and adverbs. To incorporate contextual information, features are included not only for the current sentence but for the surrounding sentences as well.

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 classifier. 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).

We adopt a conservative strategy and use the Naive Bayes classifier to label only the 90% of sentences it is most confident about. The remaining 10% are labeled as undecided

[1] Evaluated on a manually annotated test set (Wilson & Wiebe 2003), available at nrrc.mitre.org/NRRC/publications.htm, the accuracy of the system was 75%.

[2] Evaluated on a manually annotated test set (Wilson & Wiebe 2003), 82% of the objective labels and 91% of the subjective labels are correct, but recall is approximately 37% (Riloff & Wiebe 2003).

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 (Mitchell 1997) (f_i is the i^{th} feature used in the classifier):

$$CM = \mid log(\Pr(subjtive)) + \sum_i log(\Pr(f_i|subjtive)) - (log(\Pr(objtive)) + \sum_i log(\Pr(f_i|objtive))) \mid$$

The system was initially trained on a large unlabeled corpus of articles from the world press, but the resulting system was not effective in our IE experiments for the MUC-4 terrorism domain. Thus, we retrained it on the MUC-4 training set (described in the following section). The MUC-4 data has not been manually annotated with subjective/objective labels, but retraining on this corpus was possible because the system does not require labeled data.

The MUC-4 IE Task and Data

We conducted our experiments using the MUC-4 information extraction data set (MUC-4 Proceedings 1992). The MUC-4 IE task is to extract information about terrorist events. The MUC-4 corpus contains 1700 stories, mainly news articles about Latin American terrorism, and *answer key templates* containing the information that should be extracted from each story. We focused our analysis on four of the MUC-4 *string* template slots, which require textual extractions: perpetrators (individuals), victims, physical targets, and weapons. The best results reported across all string slots in MUC-4 were in the 50-70% range for recall and precision (MUC-4 Proceedings 1992).

The MUC-4 data set is divided into 1300 development (DEV) texts, and four test sets of 100 texts each (TST1, TST2, TST3, and TST4).[3] All of these texts have associated answer key templates. We used 1400 texts (DEV+TST1) as our training set, 100 texts (TST2) as a tuning set, and 200 texts (TST3+TST4) as our test set.

The IE process typically involves extracting information from individual sentences and then mapping that information into answer key templates, one template for each terrorist event described in the story. The process of template generation requires discourse processing to determine how many events took place and which facts correspond to which event. Discourse analysis is challenging even with perfect extractions, so having bad extractions in the mix makes it that much more difficult. Our goal is to use subjectivity filtering to eliminate bad extractions immediately so that the discourse processor doesn't have to grapple with them. Consequently, we evaluated the performance of our information extraction system at that stage: after extracting information from sentences, but before template generation takes place. This approach directly measures how well we are able to identify bad extractions at the stage before discourse processing would normally kick in.

[3]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.

The Information Extraction System

For this research, we created an IE system for the MUC-4 terrorism domain. To generate extraction patterns for this domain, we used the AutoSlog-TS extraction pattern learning algorithm (Riloff 1996). AutoSlog-TS requires two sets of texts for training: texts that are relevant to the domain and texts that are irrelevant to the domain. The MUC-4 data includes relevance judgements (implicit in the answer keys), so we used these judgements to partition our training set into relevant and irrelevant subsets for learning. We used the Sundance shallow parser (Riloff & Phillips 2004) to parse the documents and apply the extraction patterns.

The learning process has two steps. First, syntactic patterns are applied to the training corpus in an exhaustive fashion, so that extraction patterns are generated for (literally) every instantiation of the syntactic patterns that appears in the corpus. For example, the syntactic pattern *"<subj> passive_verb"* would generate extraction patterns for all verbs that appear in the passive voice in the corpus. The subject of the verb will be extracted. In the terrorism domain, some of these extraction patterns might be: *"<subj> was killed"*, *"<subj> was bombed"*, and *"<subj> was attacked."*

The second step applies all of the generated extraction patterns to the training corpus and gathers statistics for how often each pattern occurs in relevant versus irrelevant texts. The extraction patterns are subsequently ranked based on their association with the domain, and then human review is needed to decide which patterns to use[4] and to assign thematic roles to them. We then defined selectional restrictions for each of the four thematic roles (perpetrator, victim, target, and weapon) and automatically added these to each pattern after the reviewer assigned the thematic role.

On our training set, AutoSlog-TS generated 40,553 distinct extraction patterns. One of the authors manually reviewed all of the extraction patterns that had a score ≥ 0.951 and frequency ≥ 3. This score corresponds to AutoSlog-TS' RlogF metric, described in (Riloff 1996). The lowest ranked patterns that passed our threshold had at least 3 relevant extractions out of 5 total extractions. In all, 2,808 patterns passed this threshold and the reviewer ultimately decided that 397 of the patterns were useful for our IE task.

These 397 patterns achieved 52% recall with 42% precision on the test set.[5] These numbers are not directly comparable to the official MUC-4 scores, which evaluate template

[4]Typically, many patterns are strongly associated with the domain but will not extract information that is relevant to the IE task. For example, we only care about patterns that will extract perpetrators, victims, targets, and weapons. Some patterns may also be of dubious quality due to parsing errors.

[5]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 "5 armed men". We also discarded pronouns (they weren't scored at all) because our system does not perform coreference resolution.

System	Recall	Precision	F (β=1)	#Correct	#Wrong
(a) IE	.52	.42	.47	266	367
(b) IE+SubjFilter	.44	.44	.44	218 (-48)	273 (-94)
(c) IE+SubjFilter2	.46	.44	.45	231 (-35)	289 (-78)
(d) IE+SubjFilter2_Slct	.51	.45	.48	258 (-8)	311 (-56)
(e) IE+SubjFilter2_Slct+SubjEP	.51	.46	.48	258 (-8)	305 (-62)

Table 1: Subjectivity Filtering Results on MUC-4 Test Set

generation, but it was reassuring to see that our recall is in the same ballpark. Our precision is lower, but this is to be expected because we do not perform discourse analysis.[6]

Experiments

Row (a) of Table 1 shows the results of our IE system on the test set without any subjectivity classification. These numbers represent our baseline. The first three columns show Recall, Precision, and F-measure (β=1) scores. The last two columns show the number of correct extractions and the number of incorrect extractions.

In our first attempt at subjectivity filtering, we discarded all extractions that were found in subjective sentences. Row (b) of Table 1 shows these results. Precision increased +2% with 94 bad extractions being discarded, but recall dropped -8% because 48 correct extractions were also discarded. These results confirm that many bad extractions come from subjective sentences, but it is also clear that many good extractions are found in these sentences. We concluded that indiscriminately discarding all extractions in subjective sentences is too aggressive, because subjective language clearly can co-exist with factual information. Consequently, we decided to pursue more selective filtering strategies.

Our first modification is based on the observation that sentences with source attributions often contain factual information. News articles, in particular, often report information by citing a source (e.g., *"The Associated Press reported ..."* or *"The President stated ..."*). We observed that the presence of a source attribution in a sentence is a strong clue that the sentence contains facts that might be important to extract. Therefore we decided to override the subjectivity classifier when a sentence contains a source attribution and the sentence is not strongly subjective. So, we modified our system to override the classifier and extract information from sentences that satisfy the following two criteria: (1) the confidence measure, CM, is $\leq$ 25, indicating that the classifier considers the sentence to be only weakly subjective, and (2) the sentence contains any of the following communication verbs: {*affirm, announce, cite, confirm, convey, disclose, report, tell, say, state*}. Row (c) of Table 1 shows the results of the modified system (IE+SubjFilter2). Extracting information from the source attribution sentences improved recall by 2%, while maintaining the same level of precision.

[6]Among other things, discourse processing merges seemingly disparate extractions based on coreference resolution (e.g., 'the guerrillas" may refer to the same people as 'the armed men") and applies task-specific constraints (e.g., the MUC-4 task definition has detailed rules about exactly what types of people are considered to be terrorists).

Our second modification is aimed at being more selective about which extractions we discard. 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. We concluded that some *indicator* extraction patterns 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 give-away 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, *non-indicator* patterns represent expressions that may or may not extract relevant information. For example, *"<NP> was arrested"* and *"attributed to <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 try to automatically distinguish these two types of extraction patterns, we used the statistics generated by AutoSlog-TS on the training set. If a pattern has a conditional probability $P(relevant \mid pattern_i) \geq .65$ and a frequency $\geq$ 10, then we label it as an *indicator* pattern because it is highly correlated with the domain. Otherwise, we label the pattern as a *non-indicator pattern*. We conducted an experiment to see if the indicator patterns alone would be sufficient for our IE task. Using only the indicator patterns in our baseline system, recall dropped from 52% to 40%, demonstrating that the non-indicator patterns do extract a lot of relevant information and are important to use.

Next, we modified our system to perform *selective subjectivity filtering*: extractions from indicator patterns are never discarded, but extractions from non-indicator patterns are discarded if they appear in a subjective sentence. Row (d) of Table 1 shows the results of this selective filtering strategy, which had a dramatic impact on performance. This strategy gained an additional 5% recall, recovering 27 correct extractions that were previously discarded, while slightly increasing precision as well.

Applying Subjectivity Filtering to Objective Sentences

Our extraction patterns were manually reviewed and therefore should be of high quality, but anticipating which patterns will perform well is difficult for people because it is hard to anticipate all the ways that an expression may be used. So we wondered whether subjectivity analysis also could help us re-evaluate our extraction patterns and determine whether any of them are less reliable than we thought.

To investigate this idea, we applied both our subjectivity classifier and our extraction patterns to the training set and counted the number of times each pattern occurred in subjective vs. objective sentences. Then for each extraction pattern, we computed a probability estimate that a sentence is subjective given that it contains that pattern. We deemed an extraction pattern to be subjective if $P(subj \mid pattern_i) >$ .50 and its frequency ≥ 10.[7] These thresholds identified 10 non-indicator extraction patterns that were correlated with subjectivity:

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 that it is difficult to predict which usage will be more common. Our human reviewer expected this pattern to reliably extract targets (e.g., *"One attack was aimed at fuel storage tanks."*), but the statistics revealed that 58% of the time this expression occurs in subjective contexts, reflecting a more general use of the expression (e.g., *"The proposal is aimed at circumventing the skepticism of the Board."*).

Identifying these subjective patterns allowed us to experiment with selectively filtering subjective extractions from objective sentences. We modified our IE system to filter extractions from objective sentences if they came from any of these 10 subjective patterns.[8] Row (e) of Table 1 shows the results. This process filtered 6 additional extractions, all of which were incorrect. Although the precision increase is small, using automated subjectivity classifications to re-evaluate manually reviewed patterns costs nothing and adds more quality control to the IE process.

Our final IE system with subjectivity filtering produced a precision gain of +4% over the baseline IE system, with minimal recall loss (-1%). In absolute terms, the filtered system produced 62 fewer incorrect extractions while losing only 8 correct extractions. Table 2 breaks down the individual results for the four types of extracted information. Subjectivity filtering improved performance in all cases, increasing precision by as much as +5% on two of the four categories.

Combining Subjectivity Classification with Topic Classification

As we mentioned earlier, the MUC-4 corpus is a mixture of relevant (on-topic) texts and irrelevant (off-topic) texts that do not contain any terrorist event descriptions. So we wondered whether subjectivity filtering was eliminating bad extractions primarily from the irrelevant texts. If the filtered extractions were primarily from the irrelevant texts, then a good topic-based classifier would suffice and eliminate the need for subjectivity filtering.

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 3 shows the results of applying our baseline IE system only to the relevant texts in our test set. Precision increases by +11% compared to the results over the entire test set. This shows that many bad extractions were eliminated by removing the off-topic texts. However, the second row of Table 3 shows the results of applying our IE system with subjectivity filtering only to the relevant texts. Precision improves by +3% over the baseline system, which is almost the same level of improvement that we saw on the complete test set. These results demonstrate that subjectivity filtering is indeed eliminating bad extractions from relevant (on-topic) documents. 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.

	Baseline		**Subj Filter**	
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

Table 2: Results for Individual Slots

System	**Rec**	**Prec**
IE	.52	.53
IE+SubjFilter2_Slct+SubjEP	.51	.56

Table 3: IE Results on the Relevant Texts Only

Examples

To illustrate the behavior of the system, here we show several sentences that were classified as subjective and the extractions that were filtered as a result. The sentences are indeed subjective. Sentence (f) refers to a verbal attack, implying negative evaluation on the part of the attacker. Sentence (g) is from a speech in which the speaker is painting an adversary in a negative light (his action was a "crime" and was intended to destroy democracy). Sentence (h) describes an opinion about how to address the war on drugs. In all three cases, the extractions were correctly discarded.

(f) *The demonstrators, convoked by the solidarity with Latin America Committee, verbally attacked Salvadoran President Alfredo Cristiani and have asked the Spanish government to offer itself as a mediator to pro-*

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

[8] These extractions were already being filtered from the subjective sentences because they are non-indicator patterns.

mote an end the armed conflict.

PATTERN: attacked <dobj>
VICTIM = "Salvadoran President Alfredo Cristiani"

(g) *The crime was directed at hindering the development of the electoral process and destroying the reconciliation process ...*

PATTERN: destroying <dobj>
TARGET = "the reconciliation process"

(h) *Presidents, political and social figures of the continent have said that the solution is not based on the destruction of a native plant but in active fight against drug consumption.*

PATTERN: destruction of <np>
TARGET = "a native plant"

Conclusions

This paper presents strategies for subjectivity filtering that lead to improvements in IE performance. We also show that topic-based classification and subjectivity filtering are complementary methods for improving performance. Furthermore, automatic subjectivity classification is not yet perfect, and the system fails to identify some subjective sentences that contain bad extractions. As the coverage of subjectivity classifiers improves, we can expect further benefits for information extraction.

Acknowledgments

This work was supported in part by the National Science Foundation under grants IIS-0208798 and IIS-0208985 and the Advanced Research and Development Activity (ARDA).

References

Califf, M. E., and Mooney, R. J. 1997. Relational Learning of Pattern-Match Rules for Information Extraction. In *Proceedings of the ACL Workshop on Natural Language Learning*, 9–15.

Chieu, H.; Ng, H.; and Lee, Y. 2003. Closing the Gap: Learning-Based Information Extraction Rivaling Knowledge-Engineering Methods. In *ACL-03*.

Ciravegna, F. 2001. Adaptive Information Extraction from Text by Rule Induction and Generalisation. In *IJCAI-2001*.

Dave, K.; Lawrence, S.; and Pennock, D. M. 2003. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. In *WWW-03*.

Freitag, D., and McCallum, A. 2000. Information Extraction with HMM Structures Learned by Stochastic Optimization. In *AAAI-2000*.

Freitag, D. 1998. Toward General-Purpose Learning for Information Extraction. In *ACL-98*.

Grishman, R.; Huttunen, S.; and Yangarber, R. 2002. Information Extraction for Enhanced Access to Disease Outbreak Reports. *Journal of Biomedical Informatics* 35(4):236–246.

Hu, M., and Liu, B. 2004. Mining and Summarizing Customer Reviews. In *KDD-04*.

Klebanov, B.; Knight, K.; and Marcu, D. 2004. Text Simplification for Information Seeking Applications On the Move to Meaningful Internet Systems. *Lecture Notes in Computer Science, Springer-Verlag* 3290:735–747.

Mitchell, T. 1997. *Machine Learning*. McGraw-Hill.

Morinaga, S.; Yamanishi, K.; Tateishi, K.; and Fukushima, T. 2002. Mining Product Reputations on the Web. In *KDD-02*.

MUC-4 Proceedings. 1992. *Proceedings of the Fourth Message Understanding Conference (MUC-4)*.

Pang, B., and Lee, L. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *ACL-04*.

Riloff, E., and Phillips, W. 2004. An Introduction to the Sundance and AutoSlog Systems. Technical Report UUCS-04-015, School of Computing, University of Utah.

Riloff, E., and Wiebe, J. 2003. Learning Extraction Patterns for Subjective Expressions. In *EMNLP-03*.

Riloff, E. 1996. Automatically Generating Extraction Patterns from Untagged Text. In *AAAI-96*.

Soderland, S.; Fisher, D.; Aseltine, J.; and Lehnert, W. 1995. CRYSTAL: Inducing a Conceptual Dictionary. In *IJCAI-95*.

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

Spertus, E. 1997. Smokey: Automatic Recognition of Hostile Messages. In *IAAI-97*.

Tong, R. 2001. An Operational System for Detecting and Tracking Opinions in On-line Discussions. In *SIGIR Workshop on Operational Text Classification*.

Turney, P., and Littman, M. 2003. Measuring Praise and Criticism: Inference of Semantic Orientation from Association. *ACM Transactions on Information Systems (TOIS)* 21(4):315–346.

Wiebe, J., and Riloff, E. 2005. Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. In *CICLing-05*.

Wiebe, J.; Wilson, T.; Bruce, R.; Bell, M.; and Martin, M. 2004. Learning Subjective Language. *Computational Linguistics* 30(3):277–308.

Wilson, T., and Wiebe, J. 2003. Annotating Opinions in the World Press. In *SIGdial-03*.

Yangarber, R.; Grishman, R.; Tapanainen, P.; and Huttunen, S. 2000. Automatic Acquisiton of Domain Knowledge for Information Extraction. In *COLING-00*.

Yi, J.; Nasukawa, T.; Bunescu, R.; and Niblack, W. 2003. Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques. In *ICDM-03*.

Yu, H., and Hatzivassiloglou, V. 2003. Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. In *EMNLP-03*.