Automatic Annotation of Speech Events and Explicit Private State in Newswire

M. Arthur Munson
Cornell University
mmunson@cs.cornell.edu

May 17, 2004

Abstract

Expanding on preliminary results presented by Wiebe *et al* [18, 19], we investigate the feasibility of automatically annotating perspective and private state expressions in news text. A variety of supervised machine learning algorithms are used to identify single word annotations. The results reinforce the findings by Wiebe *et al* that the annotations are learnable. Further work is needed to include multi-word annotations in the learning framework and to use more informative learning features.

1 Introduction

Recently natural language processing (NLP) research has begun to focus on questions of subjectivity in text. While extensive previous work has studied tasks involving factual texts such as newspapers, much less is known about leveraging point of view, emotion, opinion, and subjectivity in text to aid in information retrieval, question answering, and summarization. Unsurprisingly, a large obstacle to exploring these issues is the relative lack of reliably labeled training data. In 2002 a NRRC workshop [18, 19] investigated questions of perspective in question answering tasks. One outcome from this workshop was the MPQA corpus [10], a collection of news articles with subjectivity annotations. Current work is exploring how these annotations can be used to improve natural language tasks.

As of this writing the MPQA corpus contains 535 manually annotated documents consisting of 10,000 total sentences. While this is a sufficient amount of data to conduct subjectivity research, automating annotation is a priority to enable adding domains in the future. Any research that uses the annotations in the MPQA corpus will have limited applicability unless such annotations can be easily acquired for new documents. Manually annotating opinions, as was done during the workshop, is time consuming and requires some annotator training (approximately 1-2 days in the author's personal experience). Automating just the identification of perspective and private state expressions—without attaching feature information (see Section 2)—could greatly speed further annotations. Even without the complete feature information for automatically identified annotations, a large marked corpus is potentially interesting. The presence of annotations in a text might be useful in judging the text as subjective or not. The frequency of annotations in a text might be used as a metric for how subjective the text is.

This paper reports on efforts to use supervised machine learning to identify single word perspective and private state expressions. These experiments build on a previous experiment, conducted at the end of the NRRC workshop, by using more documents and different machine learning algorithms.

The rest of the paper is organized as follows. Section 2 describes the opinion annotations that are the subject of this paper. Section 3 presents the features used to train the machine learning algorithms. The results of our experiments are given in Section 4. Related work is described in Section 5. Section 6 summarizes our conclusions.

2 Annotations

The MPQA corpus consists of newswire documents annotated with opinion information. The rest of this section briefly describes these annotations. A more complete description can be found in the NRRC MPQA workshop report [18]. Complete annotation instructions are given by Wiebe [17].

The MPQA annotations mark **ons**, **expressive-subjective elements**, and **agents**:

Ons encompass two distinct concepts. First, an on can be a **speech event**. Alternatively, an on may be an explicit acknowledgment of the **private state** of an agent. It is best to think of **on** as a shorthand to refer to either of these separate concepts.

Speech events are simply acts of communication.

Private state refers to the opinions, emotions, and beliefs of an agent. While these are not physically observable and are therefore not public knowledge, they are necessary to fully describe and understand that individual.

An agent is the source or target of an opinion or speech event.

Expressive-subjective elements are phrases that imply opinions and subjectivity. They are usually idiomatic expressions or words with strong connotations.

Figure 1 shows an annotated document from the MPQA corpus that contains both ons and expressive-subjective elements. While most of the ons are speech events (*e.g.* "reports", "saying", "asked", "invited"), the on [expressed alarm]$_{on}$ is arguably both a speech event and an explicit acknowledgment of the Premier's private state.

Additional information is attached to each MPQA annotation as features. The last sentence in Figure 1 shows this information as `XML` attributes. This information includes the source of ons and expressive-subjective elements, the strength of opinions, the confidence of the annotator in making the annotation, and other information.

A few important points about agents are worth mentioning. First, agent annotations have a nested structure. For example, [the police]$_{agent}$ in the last sentence is a nested source beneath the implied writer agent (abbreviated as `w`). In sentence four, [the gangsters]$_{agent}$ is nested below [Huang]$_{agent}$ (who is in turn nested below [writer]$_{agent}$). Second, the agent annotations explicitly resolve noun phrase coreferences by using the `id` specified in the agent's first appearance. This can be seen in the last sentence where [they]$_{agent}$ has the nested source `w, police`. Third, agents are not restricted to people. The second to last sentence includes [press reports]$_{agent}$ as the source agent for "... [said]$_{on}$ he had ..."

It is important to note that not all ons correspond to subjectivity in the text. In the last sentence both occurrences of [said]$_{on}$ are `onlyfactive`, meaning that the on is objective. The ons [expressed alarm]$_{on}$, [claimed]$_{on}$, and [demanded]$_{on}$ are examples of ons in the example that are not `onlyfactive`.

The work presented here only tries to identify the location of ons in documents. Automatically filling in the complete feature information for an on annotation is a complicated task that is only interesting if ons can be reliably identified using automatic methods. Expressive-subjective elements are not considered because inter-annotator agreement for these annotations is too low to expect a computer to do well.[1]

As stated in Section 1, our experiments only dealt with single word ons. This restriction is a result of the machine learning formulation (see Section 3). Table 1 shows the distribution of ons based on the number of words in each on. While the majority of ons consist of a single word (58%), a large portion of ons are multi-word ons. The very long ons are somewhat surprising; we have not yet investigated whether these are annotation errors.

3 Training Features

Following the proof of concept experiment reported by Wiebe *et al* [18, 19], each training instance corresponds to a token in a document. A token is either a word or a piece of punctuation (*e.g.* period, comma). For

[1] Wiebe *et al* [18, 19] report that agreement for expressive-subjective elements was 69%, compared to 85% for ons.

Taipei, Jan. 3 (CNA) –

[Premier Chang Chun-hsiung]$_{agent}$ [expressed alarm]$_{on}$ Thursday at [reports]$_{on}$ that a legislator was kidnapped and held for three days in a Taipei hotel, [saying]$_{on}$ that his government [will do its utmost]$_{es}$ to stop gangsters from [preying]$_{es}$ on lawmakers.

Presiding over an inter-agency conference on social order, [Chang]$_{agent}$ [asked]$_{on}$ the police to catch the gangsters who allegedly detained Legislator [Huang Hsien-chou]$_{agent}$ at the Grand Hyatt Hotel in Taipei.

Later that day, a man was arrested in Taipei in relation to the case. Chang Fu-shun, 27, [is said]$_{on}$ to be the younger brother of [Chang Hui-hua]$_{agent}$, who allegedly [invited]$_{on}$ Huang to meet her at the hotel and who [Huang]$_{agent}$ [claimed]$_{on}$ was in the hotel room when he was detained.

[Huang]$_{agent}$ [claimed]$_{on}$ the previous day in Taichung that he had been trapped by [the gangsters]$_{agent}$, who [had pretended]$_{on}$ to be his political supporters.

[Huang]$_{agent}$ [told]$_{on}$ reporters that the gangsters, who included at least two men and a woman, made an appointment to meet him at the hotel Dec. 26. When he arrived, [he]$_{agent}$ [claimed]$_{on}$, one of them offered him a drink and he lost consciousness after drinking it.

When he awoke, he found that his hands and feet had been bound. [The gangsters]$_{agent}$ [demanded]$_{on}$ NT$2 million from him but finally [settled]$_{on}$ for NT$800,000 which they withdrew using Huang's credit card.

Huang was released Dec. 31 and reported to police that day.

[Huang]$_{agent}$ [revealed]$_{on}$ what [he]$_{agent}$ [claimed]$_{on}$ to be the whole story at a news conference Wednesday night after [press reports]$_{agent}$ [said]$_{on}$ he had [in fact]$_{es}$ been caught with prostitutes in the hotel and blackmailed.

<agent nested-source=''w, police'' id=''police''>The police</agent> <on nested-source=''w, police'' onlyfactive=''yes''>said</on> that the hotel room had been paid for by a woman who checked in under the name of Lin An, which <agent nested-source=''w, police''>they</agent> <on nested-source=''w, police'' onlyfactive=''yes''>said</on> was an alias used by Chang Hui-hua.

Figure 1: Annotations for document 20020103/20.35.17-19238 in MPQA corpus. Annotation features have been omitted for all but the last sentence to improve legibility. In these sentences **ons** are marked by [ ]$_{on}$, **agents** by [ ]$_{agent}$, and **expressive-subjective** elements by [ ]$_{es}$. The annotations in the last sentence use `XML` formatting and include the detailed feature information.

Length of Ons

No. Words	Count	% of All Ons
1	7252	58.0346%
2	2912	23.3035%
3	1318	10.5474%
4	580	4.64149%
5	230	1.84059%
6	124	0.992318%
7	46	0.368118%
8	13	0.104033%
9	7	0.0560179%
10	6	0.0480154%
11	6	0.0480154%
12	0	0.0%
13	1	0.00800256%
14	1	0.00800256%

Table 1: Distribution of ons by their length, measured in words. *Count* is the number of times an on of a given length occurred in all 535 documents; *% of All Ons* is the percentage of ons with the given length.

clarity, **instance-word** below refers to the token corresponding to the instance. The following features are generated for each instance:

- Three features (`a1 - a3`) generated by `CASS` [1] that are the `CASS` names for a series of (collapsed) parse chunks (*e.g.* `vb`, `cc`, `in`, `nn`). The first is the chunk with a span matching the instance-word. The second and third correspond to the previous and successor chunks, respectively.
- Features `a4 - a8` form a lexical context window around the instance-word. Feature `a6` is the word (token) corresponding to the instance; the other features are the two preceding and two successive tokens.[2] None of the words are stemmed.
- Feature `a9` is the part of speech for the instance-word.
- Feature `a10` is the category of the instance-word's lemma on a word list. The word list is designed to include likely speech event and private state words. If the instance-word is not on the list this feature is set to `NO`. The word list used is taken from Levin [8] and Framenet [5].

The learning task is to classify an instance as an on (`YESon`) or not an on (`NOon`).

4 Experimental Results

Training instances were generated for all the tokens in 400 documents. The combined set of tokens resulted in approximately 16,000 possible values for each of the five lexical features (`a4 - a8`). The results below represent averages from using ten fold cross-validation. Each fold consisted of roughly 200,000 instances in the training set and approximately 20,000 instances in the test set. Exact sizes varied since folds were taken at the document level, not the token level, with 40 documents in the test set and 360 documents in the training set. Each training and test set contained about 4,700 and 500 ons, respectively.

Performance is measured by the precision, recall, and F-measure for the `YESon` class. Accuracy is a poor metric for this task since the class distribution is extremely skewed. Simply choosing `NOon` yields very high

[2]Due to the script implementation that generates the window, the words are in reverse order. That is, `a4` is the second word after the instance-word, and `a8` is the word two before the instance-word.

accuracy. If G is the set of ons in the gold standard (the "true" set of ons), and S is the set of ons annotated by the classifier (the predicted set of ons), the metrics can be defined formally as:

$$Precision = \frac{|G \cap S|}{|S|} \tag{1}$$

$$Recall = \frac{|G \cap S|}{|G|} \tag{2}$$

$$F - Measure = \frac{2|G||S|}{|G| + |S|} \tag{3}$$

We evaluated the performance of four different machine learning systems on the data: `TiMBL` [4], `SVM`^{light} [7], `RIPPER` [3], and `C4.5` [12]. `TiMBL` is an implementation of k-nearest neighbors. The results below were obtained for k=1. `SVM`^{light} is an efficient support vector machine (SVM) implementation. `RIPPER` is a rule learner that searches for hypotheses that can classify instances. `C4.5` is a decision tree implementation. All systems were trained and then tested on the unseen data for the fold. Default parameter values were used in all cases.

The baseline we used was from the word list feature (`a10`). If the instance-word's lemma had a category of either `fn_communication_statement_se_v` or `bl_str_se_verb` the instance was classified as an on. These categorizations are taken from Framenet's [5] communication domain and Section 37.7 of *English Verb Classes and Alternations* [8], respectively. These categories were chosen because they contain "say", which is always a speech event. Note that this is the same baseline used by Wiebe *et al* [18, 19]. Table 2 summarizes the results of our experiments and the results of the original learning experiment reported by Wiebe *et al*.

Performance Results for Tagging Ons

	Precision	**Recall**	**F-Measure**
`TiMBL` [4] (k=1)	0.7453 ±0.0258	*0.6532* ±0.0314	*0.6961* ±0.0285
`SVM`^{light} [7]	*0.8865* ±0.0198	0.5523 ±0.0441	0.6800 ±0.0383
`RIPPER` [3]	0.8013 ±0.0318	0.5337 ±0.0328	0.6404 ±0.0319
`C4.5` [12]	0.7793 ±0.0484	0.4279 ±0.0448	0.5505 ±0.0350
Baseline	0.7541 ±0.0187	0.4111 ±0.0315	0.5315 ±0.0270
MPQA Workshop Results [18, 19]			
k-NN (k=1)	0.6960	0.6340	0.6640
Naïve Bayes	0.4670	0.7660	0.5800
Baseline	0.6990	0.4770	0.5670

Table 2: Results for identifying single word ons. The top portion of the table shows averages obtained using 10-fold cross-validation for 400 documents. Standard deviations are given. The highest value in each column (with respect to the top of the table) is *italicized*. The bottom of the table reproduces the results obtained by Wiebe *et al* [18, 19] using 10-fold cross-validation over 100 documents. The baseline was the same for both experiments.

Several observations can be made from Table 2. First, k-nearest neighbor shows the best recall and overall performance, according to F-measure, for both our experiments and those reported previously. Using more documents seems to improve performance; `TiMBL` gets a higher precision and shows a slight improvement in recall over the results using 100 documents. Some of this difference may be due to using a different implementation than in the original experiment.

With a slightly lower F-measure, `SVM`^{light} gets a much higher precision. This suggests that a SVM is a viable option for automatically annotating the "easy" ons. Subsequent passes through the data by human annotators would be needed to find the remaining ons.

`RIPPER` has the second highest precision and slightly lower recall than `SVM`^{light}. `C4.5` shows similar behavior to `RIPPER`, with slightly lower precision and much lower recall. While these algorithms do not perform as well as `TiMBL` or `SVM`^{light}, they are still useful tools for evaluating the utility of the training features. Section 4.1 looks at the models generated by `C4.5` and `RIPPER`.

Unsurprisingly, the baseline has a high precision but low recall. In fact, the precision is competitive with C4.5 and TiMBL. On the larger set of documents the baseline performs slightly worse overall than on the smaller set of documents. This is likely related to the larger set of words. As a result, k-nearest neighbor has an F-measure a full 16 points higher than the baseline.

4.1 Model Analysis

It is interesting to look at the models produced by the different algorithms to gain insight into their behavior. While it is not possible to look at the models for k-nearest neighbor or SVM, we can examine C4.5's trees and RIPPER's rules. The tree and rule samples below were both produced for the first fold of the data.

```
format: attr = value: class
        a10 = NO: NOon (198133.0/1668.7)
        a10 = bl_wk_se_verb: NOon (3213.0/608.9)
        a10 = fn_communication_request_se_n: NOon (165.0/32.0)
        a10 = fn_communication_encoding_se_v: NOon (174.0/18.3)
        a10 = fn_communication_statement_se_n: NOon (528.0/174.0)
        a10 = fn_communication_questioning_se_n: NOon (81.0/18.1)
        a10 = fn_communication_statement_se_v: YESon (2237.0/467.8)
        a10 = fn_communication_request_se_v: YESon (179.0/74.1)
        a10 = fn_communication_conversation_se_v: NOon (127.0/42.2)
        a10 = fn_communication_commitment_se_n: NOon (71.0/20.1)
        a10 = fn_communication_conversation_se_n: NOon (153.0/35.1)
        a10 = fn_cognition_cogitation_se_v: NOon (20.0/2.5)
        a10 = fn_communication_communication_response_se_n: NOon (54.0/14.8)
        a10 = bl_mod_se_verb: NOon (130.0/56.4)
        a10 = fn_communication_hear_se_v: NOon (43.0/7.2)
        a10 = fn_communication_questioning_se_v: NOon (22.0/13.1)
        a10 = fn_communication_candidness_se_a: NOon (33.0/1.4)
        a10 = fn_communication_encoding_se_n: NOon (15.0/4.7)
        a10 = fn_communication_manner_se_v: YESon (8.0/4.5)
        a10 = fn_communication_gesture_se_v: NOon (5.0/1.2)
        a10 = fn_communication_volubility_se_a: NOon (1.0/0.8)
        a10 = fn_communication_communication_noise_se_v: NOon (5.0/2.3)
        a10 = fn_body_body-movement_se_v: NOon (1.0/0.8)
        a10 = fn_cognition_invention_se_v: NOon (1.0/0.8)
        a10 = fn_communication_communication_response_se_v:
        a10 = bl_str_se_verb:
        a10 = fn_cognition_judgment_se_v:
        a10 = fn_communication_commitment_se_v:
```

Figure 2: The pruned decision tree for one fold of the data. Only the top-most decision branch is shown. Lines that end with a class prediction are leaves; otherwise further attribute values are checked before making a prediction (not shown). The numbers in parentheses show the correct-incorrect ratio for leaves *over the training set.*

Figure 2 displays the pruned decision tree for fold 1. C4.5 relies heavily on feature a10, the list of word categories, to decide if an instance is an on. Very few categories require the decision tree to test other features. Of the roughly 4,700 ons in the training set, less than 300 fall into the categories with deeper paths to the leaves. The two largest leaves that correctly classify ons are fn_communication_statement_se_v and fn_communication_request_se_v. The first category contains speech event verbs that are communication statements (*e.g.* add, address, admit, declare, gripe, lecture, recount, say); the second contains speech event verbs that are communication requests (*e.g.* ask, beg, command, tell, urge). It is not surprising that these categories are strong indicators of ons. On the other hand, it is surprising how often words in these categories *are not* ons. Between the two categories, nearly 500 words are falsely classified as ons. From the words in

the categories, a likely explanation is that many words have multiple uses. *Add* and *lecture*, for example, both have other meanings that have no relation to speech events. In some cases, such as *lecture*, `C4.5` could have distinguished between different meanings by looking at the word's part of speech. In others, such as *add*, more context would have needed to be taken into consideration. Ultimately `C4.5` is impaired by the skew of the data; the distribution appears to cause heavy pruning that yields a very "stump"-like tree.

```
format: class :- precondition(s)
     1: YESon :- a6=says (117/4).
     2: YESon :- a6=added (57/3).
     3: YESon :- a6=said (1047/58).
     4: YESon :- a6=believes (18/0).
     5: YESon :- a6=told (139/9).
     6: YESon :- a6=thinks (12/0).
     7: YESon :- a6=insisted (11/1).
     8: YESon :- a6=noted (27/2).
    38: YESon :- a6=supported, a9=VBD (14/0).
    45: YESon :- a6=warned, a9=VBD (25/1).
    71: YESon :- a6=announced, a1=vx (38/2).
    78: YESon :- a6=blamed, a1=vx (8/0).
    96: YESon :- a6=believe, a9=VBP (32/2).
    99: YESon :- a6=saying, a7=P_COMMA (46/0).
   112: YESon :- a6=quoted, a1=vx (7/0).
   121: YESon :- a10=fn_communication_manner_se_v (5/3).
   130: YESon :- a6=cited, a9=VBD (10/2).
   131: YESon :- a6=wish, a9=VBP (7/4).
   143: YESon :- a4=RegistrarP_HYPHEN (2/1).
   145: YESon :- a6=dismissed, a1=vx (5/0).
   147: YESon :- a6=suggested, a1=vx (6/2).
   149: YESon :- a6=ordered, a9=VBD (5/1).
   168: YESon :- a6=described, a9=VBD (22/2).
   194: YESon :- a6=declared, a3=vx (19/0).
   199: YESon :- a6=predicted, a3=vx (5/0).
   201: YESon :- a6=sought, a1=vx (5/0).
   205: YESon :- a6=accused, a9=VBD (22/2).
   208: YESon :- a7=Congolese (2/0).
   209: YESon :- a7=Ukrainians (2/0).
   222: YESon :- a6=know, a9=VBP (23/3).
   223: YESon :- a6=rejected, a1=vx (9/0).
```

Figure 3: Selected rules generated by `RIPPER` [3] for classifying words as ons. Each rule is prefixed with its precedence. To use a rule to classify an instance, the instance must meet the rule's precondition(s). A precondition constrains an attribute (*e.g.* `a6`) to a particular value (*e.g.* `says`). The numbers in parentheses give the correct-incorrect ratio for the rule when applied to a test set.

In contrast, `RIPPER` almost exclusively looks at the instance-word itself (feature `a6`) to decide if it is an on. Figure 3 shows a sampling of the rules generated for fold 1. Almost all of the rules not shown test `a6` alone for a particular value. In some sense, `RIPPER` constructs its own word list. Oddly, only one rule tests the word list feature (`a10`).

Unsurprisingly, rules involving *says* and *said* are very high on the rule list. On the other hand, both rules misclassify some instances as ons. Given that *to say* epitomizes speech events, these misclassifications suggest that there are some annotations missing in the corpus. We have not yet examined the individual misclassifications.

In a few rules part of speech (feature `a9`) and `CASS` parse information (features `a1-a3`) are used with the instance-word to help classify ons. With respect to the parse chunks, `RIPPER` mostly relies on the chunk

containing the instance-word. From the rules it is impossible to tell how much testing these additional features improves performance.

Finally, `RIPPER` does not use the lexical context of the instance hardly at all. Only rule 99 involves the lexical context and is reasonable. Rule 99 corresponds to the construction "..., saying ...". The other rules testing features `a4-a5` and `a7-a8` appear to be overfitted to the data. Rules 143, 208, and 209 do not logically have any relation to speech events or to private state. Trials on the other folds exhibit similar tendencies, with most of the context-using rules overfitting to the data.

5 Related Work

The work by Wiebe *et al* [18, 19] done for the NRRC MPQA workshop is obviously very closely related to this work. While the same learning experiment was run, this work used a larger set of documents and explored different machine learning algorithms. More significantly, the time constraints of the workshop prevented Wiebe *et al* from analyzing how the training features were used by learning algorithms. Our work has taken a small but important extra step in investigating the problem of automatically annotating speech events and explicit private state expressions.

A fair amount of research has explored various clues that indicate subjectivity in text. These clues could potentially be incorporated into future learning experiments as features. Hatzivassiloglou and McKeown [6] presented a way to automatically determine the semantic orientation of conjoined adjectives. Also working with adjectives, Wiebe [16] investigated learning subjective adjectives from corpora starting from a small set of strongly subjective adjectives. Working with the MPQA corpus, Riloff *et al* [13, 14] used information extraction patterns and multiple levels of bootstrapping to find subjective nouns and subjective expressions. One problem with automatically identified subjective expressions is that the extracted expressions are noisy; not all the expressions are truly subjective. To reduce the need to manually sort through candidate expressions, Wiebe and Wilson [20] used the density of nearby subjective-elements (*e.g.* subjective adjectives and verbs, unique words) to improve the precision of potentially subjective expressions identified using n-grams as proposed by Wiebe, Wilson, and Bell [21].

Older research by Bergler [2] characterized the lexical semantics of seven reporting verbs (*e.g.* to say) in a lexicon to support distinguishing between primary information (the facts being reported) and the circumstantial information framing the facts. The task of identifying speech events would be greatly simplified given a complete reporting verb lexicon. Conversely, understanding how to correctly identify speech events could translate to being able to automatically construct such a lexicon.

As `C4.5` illustrated in our experiments, word categories are a potentially powerful tool to finding private state expressions and speech events. They can also be coarse grained. One solution might be to build a fuzzy lexicon as proposed by Subasic and Huettner [15]. Instead of encoding the affect categories of a word and the intensity with which the word represents those categories, the lexicon could contain the word classification categories used in our experiments. The key point would be using fuzzy logic to create groups of similarly grouped words. Instead of using a discrete category as a learning feature, a vector of category features could be used for learning. Each category feature would be a continuous value indicating the affinity of the word to the group. Using such a continuous scale would reduce the coarseness of word categories and possibly improve automatic annotation.

At a higher level, a recent area of subjectivity that has interested researchers is labeling entire sentences and documents as subjective or objective with the aim of improving information retrieval and question answering systems. Liu *et al* [9] and Yu and Hatzivassiloglou [22] have approached this problem using knowledge-based and statistical methods, respectively. The presence of ons may be a good indicator of subjectivity and could be easily incorporated into the statistical framework used by Yu and Hatzivassiloglou.

With respect to subjective sentences and documents, research by Pang *et al* [11] and Yu and Hatzivassiloglou [22] has tried to further classify text as positive or negative. Pang *et al* report that document sentiment classification for movie reviews is hindered by digressions that recount the plot and characters in the movie. Identifying ons that mark perspective changes may be helpful to sentiment analysis by removing noise.

6 Conclusions

This paper has explored the question of whether existing machine learning algorithms are able to automatically annotate single word perspective and private state expressions (*ons*) in news documents. The results confirm the findings of a previous learning experiment conducted by Wiebe *et al* [18, 19] and show that it is feasible. K-nearest neighbors showed the best overall performance (0.6961 F-measure) , while a support vector machine showed the best precision (0.8865 precision).

More importantly, none of the tested systems demonstrated exceptional performance, suggesting that future work needs to be done to improve the set of learning features. Preliminary analysis of the generated decision tree and rule learning models indicates that neither system is effectively using the context around the instance. It is not clear if using context would greatly improve performance; it may be that context information is far less important than categorizing words as likely ons.

Finally, future work is needed to explore alternative formulations that permit learning multi-word expressions.

Acknowledgments Eric Breck provided invaluable assistance by working with a dozen or so cryptic scripts to generate the supplementary annotations needed to construct the instances. Without his help I would likely still be puzzling through the code written during the NRRC MPQA workshop.

References

[1] S. Abney. Rapid incremental parsing with repair. In *Proceedings fo the 6th New OED Conference*, University of Waterloo, Waterloo, Ontario, 1990.

[2] S. Bergler. The semantics of collocational patterns for reporting verbs. In *EACL91P*, pages 216–221, 1991.

[3] W. W. Cohen. Fast effective rule induction. In A. Prieditis and S. Russell, editors, *Proc. of the 12th International Conference on Machine Learning*, pages 115–123, Tahoe City, CA, July 9–12, 1995. Morgan Kaufmann.

[4] W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. TiMBL: Tilburg memory based learner, version 3.0, reference guide. ILK Technical Report 00-01, Tilburg University, 2000. Available from `http://ilk.kub.nl/~ilk/papers/ilk0001.ps.gz`.

[5] Framenet. See `http://www.icsi.berkeley.edu/~framenet/`.

[6] V. Hatzivassiloglou and K. R. McKeown. Predicting the semantic orientation of adjectives. In P. R. Cohen and W. Wahlster, editors, *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 174–181, Somerset, New Jersey, 1997. Association for Computational Linguistics.

[7] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, USA, 1999.

[8] B. Levin. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago, 1993.

[9] H. Liu, H. Lieberman, and T. Selker. A model of textual affect sensing using real-world knowledge. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 125–132. ACM Press, 2003.

[10] MPQA Corpus. Available from `http://nrrc.mitre.org/NRRC/Docs_Data/MPQA_04/approval_mpqa.htm`.

[11] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, 2002.

[12] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[13] E. Riloff and J. Wiebe. Learning extraction patterns for subjective expressions. In *Proc. of the 2003 Conf. on Empirical Methods in Natural Language Processing (EMNLP-03)*, 2003.

[14] E. Riloff, J. Wiebe, and T. Wilson. Learning subjective nouns using extraction pattern bootstrapping. In W. Daelemans and M. Osborne, editors, *Proceedings of CoNLL-2003*, pages 25–32. Edmonton, Canada, 2003.

[15] P. Subasic and A. Huettner. Affect analysis of text using fuzzy semantic typing. *IEEE-FS*, 9:483–496, Aug. 2001.

[16] J. Wiebe. Learning subjective adjectives from corpora. In *AAAI/IAAI*, pages 735–740, 2000.

[17] J. Wiebe. Instructions for annotating opinions in newspaper articles. Technical Report TR-02-101, University of Pittsburgh, 2002.

[18] J. Wiebe, E. Breck, C. Buckley, C. Cardie, P. Davis, B. Fraser, D. Litman, D. Pierce, E. Riloff, and T. Wilson. NRRC Summer Workshop on Multiple-Perspective Question Answering Final Report. Tech report, Northeast Regional Research Center, Bedford, MA, 2002.

[19] J. Wiebe, E. Breck, C. Buckley, C. Cardie, P. Davis, B. Fraser, D. Litman, D. Pierce, E. Riloff, T. Wilson, D. Day, and M. Maybury. Recognizing and organizing opinions expressed in the world press. In *Papers from the AAAI Spring Symposium on New Directions in Question Answering (AAAI tech report SS-03-07)*, 2003. March 24-26, 2003. Stanford University, Palo Alto, California.

[20] J. Wiebe and T. Wilson. Learning to disambiguate potentially subjective expressions. In *Proceedings of CoNLL-2002*, pages 112–118. Taipei, Taiwan, 2002.

[21] J. Wiebe, T. Wilson, and M. Bell. Identifying collocations for recognizing opinions, 2001.

[22] H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proc. of the 2003 Conf. on Empirical Methods in Natural Language Processing (EMNLP-03)*, pages 129–136, 2003.