Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques

Jeonghee Yi[†] Tetsuya Nasukawa[‡] Razvan Bunescu[⋆] [*] Wayne Niblack[†]

[†] IBM Almaden Research Center, 650 Harry Rd, San Jose, CA 95120, USA
{jeonghee, niblack}@almaden.ibm.com
[‡] IBM Tokyo Research Lab, 1623-14 Shimotsuruma, Yamato-shi, Kanagawa-ken 242-8502, Japan
nasukawa@jp.ibm.com
[⋆] Dept. of Computer Science, University of Texas, Austin, TX 78712, USA
razvan@cs.utexas.edu

Abstract

We present Sentiment Analyzer (SA) that extracts sentiment (or opinion) about a subject from online text documents. Instead of classifying the sentiment of an entire document about a subject, SA detects all references to the given subject, and determines sentiment in each of the references using natural language processing (NLP) techniques. Our sentiment analysis consists of 1) a topic specific feature term extraction, 2) sentiment extraction, and 3) (subject, sentiment) association by relationship analysis. SA utilizes two linguistic resources for the analysis: the sentiment lexicon and the sentiment pattern database. The performance of the algorithms was verified on online product review articles ("digital camera" and "music" reviews), and more general documents including general webpages and news articles.

1. Introduction

Today, a huge amount of information is available in online documents such as web pages, newsgroup postings, and on-line news databases. Among the myriad types of information available, one useful type is the *sentiment*, or *opinions* people express towards a *subject*. (A *subject* is either a topic of interest or a feature of the topic.) For example, knowing the reputation of their own or their competitors' products or brands is valuable for product development, marketing and consumer relationship management. Traditionally, companies conduct consumer surveys for this purpose. Though well-designed surveys can provide quality estimations, they can be costly especially if a large volume of survey data is gathered.

There has been extensive research on automatic text analysis for sentiment, such as sentiment classifiers[13, 6, 16, 2, 19], affect analysis[17, 21], automatic survey analysis[8, 16], opinion extraction[12], or recommender systems [18]. These methods typically try to extract the overall sentiment revealed in a document, either positive or negative, or somewhere in between.

Two challenging aspects of sentiment analysis are: First, although the overall opinion about a topic is useful, it is only a part of the information of interest. Document level sentiment classification fails to detect sentiment about individual aspects of the topic. In reality, for example, though one could be generally happy about his car, he might be dissatisfied by the engine noise. To the manufacturers, these individual weaknesses and strengths are equally important to know, or even more valuable than the overall satisfaction level of customers.

Second, the association of the extracted sentiment to a specific topic is difficult. Most statistical opinion extraction algorithms perform poorly in this respect as evidenced in [3]. They either i) assume the topic of the document is known *a priori*, or ii) simply associate the opinion to a topic term co-existing in the same context. The first approach requires a reliable $topic$ or *genre classifier* that is a difficult problem in itself. A document (or even a portion of a document as small as a sentence) may discuss multiple topics and contain sentiment about multiple topics.

For example, consider the following sentences from which *ReviewSeer*[3] found positive opinions about the `NR70` PDA:

```
1. As with every Sony PDA before it, the NR70
   series is equipped with Sony's own Memory
   Stick expansion.
```

[*] The author's work on a portion of the feature term selection algorithm development was performed while the author was on summer internship at IBM Almaden Research Center.

```
2. Unlike the more recent T series CLIEs,
   the NR70 does not require an add-on
   adapter for MP3 playback, which is
   certainly a welcome change.
3. The Memory Stick support in the NR70
   series is well implemented and functional,
   although there is still a lack of non-
   memory Memory Sticks for consumer
   consumption.
```

Based on our understanding of the *ReviewSeer* algorithm, we suppose their statistical method (and most other statistical opinion extraction methods) would assign the same polarity to `Sony PDA` and `T series CLIEs` as that of `NR70` for the first two sentences. That is wrong for `T series CLIEs`, although right for `Sony PDA`. We notice that the third sentence reveals a negative aspect of the `NR70` (i.e., the lack of non-memory Memory Sticks) as well as a positive sentiment in the primary phrase.

We anticipated the shortcomings of the purely statistical approaches, and in this paper we show that the analysis of grammatical sentence structures and phrases based on NLP techniques mitigates some of the shortcomings. We designed and developed *Sentiment Analyzer* (*SA*) that

- extracts topic-specific features
- extracts sentiment of each sentiment-bearing phrase
- makes (topic|feature, sentiment) association

SA detects, for each occurrence of a topic spot, the sentiment specifically about the topic. It produces the following output for the above sample sentences provided that `Sony PDA`, `NR70`, and `T series CLIEs` are specified topics:

```
1. Sony PDA - positive
   NR70 - positive
2. T series CLIEs - negative
   NR70 - positive
3. NR70 - positive
   NR70 - negative
```

The rest of this paper is organized as follows: Section 2 describes the feature term extraction algorithm and reports the experimental results for feature term selection. Section 3 describes the core sentiment detection algorithms and experimental results. Section 4 summarizes related work and compares them with our algorithms. Finally, we conclude with a discussion in Section 5.

2. Feature Term Extraction

A *feature term* of a topic is a term that satisfies one of the following relationships:

- a *part-of* relationship with the given topic.
- an *attribute-of* relationship with the given topic.
- an *attribute-of* relationship with a known feature of the given topic.

This **camera** has everything that you need. It takes great **pictures** and is very easy to use. It has very good **documentation**. Bought 256 MB **memory card** and can take a huge number of **pictures** at the highest **resolution**. Everyone is amazed at the **resolution** and clarity of the **pictures**. The results have been excellent from **macro shots** to **telephoto nature shots**. **Manuals** and **software** are not easy to follow. Good **Battery Life** 200 on 1GB **drive** Best **Remote** I have seen on any **camera**. The **battery** seems to last forever but you will want a spare anyway. The best **built in flash** I have seen on any camera. The G2 has enough **features** to keep the consumer and pro creative for some time to come!

Figure 1. Sample digital camera review

For the digital camera domain, a feature can be a *part of* the camera, such as `lenses, battery` or `memory card`; an *attribute*, such as `price` or `size`; or an *attribute of a feature*, such as `battery life` (an attribute of feature `battery`). Figure 1 is a portion of an actual review article from `www.cnet.com`. The phrases in bold are the features we intend to extract. We apply the feature term extraction algorithm described in the rest of this section to a set of documents having the same topic.

2.1. The Candidate Feature Term Selection

Based on the observation that feature terms are nouns, we extract only noun phrases from documents and apply feature selection algorithms described in Section 2.2. Specifically, we implemented and tested the following three candidate term selection heuristics.

2.1.1. Base Noun Phrases (BNP). *BNP* restricts the candidate feature terms to one of the following base noun phrase (BNP) patterns: *NN*, *NN NN*, *JJ NN*, *NN NN NN*, *JJ NN NN*, *JJ JJ NN*, where *NN* and *JJ* are the part-of-speech(POS) tags for nouns and adjectives respectively defined by Penn Treebank[10].

2.1.2. Definite Base Noun Phrases (dBNP). *dBNP* further restricts candidate feature terms to definite base noun phrases, which are noun phrases of the form defined in Section 2.1.1 that are preceded by the definite article "the." Given that a document is focused on a certain topic, the definite noun phrases referring to topic features do not need any additional constructs such as attached prepositional phrases or relative clauses, in order for the reader to establish their referent. Thus, the phrase "the battery," instead of "the battery of the digital camera," is sufficient to infer its referent.

2.1.3. Beginning Definite Base Noun Phrases (bBNP). *bBNP* refers to *dBNP* at the beginning of sentences followed by a verb phrase. This heuristic is based on the observation that, when the focus shifts from one feature to another, the new feature is often expressed using a definite noun phrase at the beginning of the next sentence.

2.2. Feature Selection Algorithms

We developed and tested two feature term selection algorithms based on a mixture language model and likelihood ratio. They are evaluated in Section 2.3.

2.2.1. Mixture Model. This method is based on the mixture language model by Zhai and Laffertry[23]: they assume that an observed documents d is generated by a mixture of the query model and the corpus language model. In our case, we may consider our language model as the mixture (or a linear combination) of the general web language model θ_W (similar to the corpus language model) and a topic-specific language model θ_T (similar to the query model):

$$\theta = \alpha\theta_W + \beta\theta_T$$

where α, β are given and sum to 1. α indicates the amount of background noise when generating a document from the topic-specific model. θ, θ_W and θ_T have multinomial distributions, $\theta_W = (\theta_{W_1}, \theta_{W_2}, ...\theta_{W_k})$, $\theta_T = (\theta_{T_1}, \theta_{T_2}, ...\theta_{T_k})$, and $\theta = (\theta_1, \theta_2, ...\theta_k)$, where k is the number of words in the corpus. Intuitively, by calculating the topic-specific model, θ_T, noise words can be deleted, since the topic-specific model will concentrate on words occurring frequently in topic-related documents, but less frequently in the whole corpus. The maximum likelihood estimator of θ_W can be calculated directly as:

$$\hat{\theta_{W_i}} = \frac{df_i}{\sum_j df_j}$$

where df_i is the number of times word i occurs in the whole corpus. The problem of finding θ_T can be generalized as finding the maximum likelihood estimation of multinomial distribution θ_T.

Zhang *et al.*[24] developed an $O(klog(k))$ algorithm that computes the exact maximum likelihood estimation of the multinomial distribution of q in the following mixture model of multinomial distributions, $p = (p_1, p_2, ...p_k)$, $q = (q_1, q_2, ...q_k)$, and $r = (r_1, r_2, ...r_k)$:

$$r = \alpha p + \beta q$$

Let f_i be the observed frequency of word i in the documents that are generated by r. Sort $\frac{p_i}{f_i}$ so that $\frac{f_1}{p_1} > \frac{f_2}{p_2} > ... > \frac{f_k}{p_k}$. Then, find t that satisfies:

$$\begin{aligned} \frac{\frac{\beta}{\alpha} + \sum_{j=1}^{t} p_j}{\sum_{j=1}^{t} f_j} - \frac{p_t}{f_t} &> 0 \\ \frac{\frac{\beta}{\alpha} + \sum_{j=1}^{t+1} p_j}{\sum_{j=1}^{t+1} f_j} - \frac{p_{t+1}}{f_{t+1}} &\leq 0 \end{aligned}$$

	D_+	D_-
bnp	C_{11}	C_{12}
$\overline{bnp}$	C_{21}	C_{22}

Table 1. Counts for a *bnp* [9]

Then, the q_i's are given by:

$$\begin{aligned} q_i &= \begin{cases} \frac{f_i}{\lambda} - \frac{\alpha}{\beta} p_i & \text{if } 1 \leq i \leq t \\ 0 & \text{otherwise} \end{cases} \\ \lambda &= \frac{\sum_{i=1}^{t} f_i}{1 + \frac{\alpha}{\beta} \sum_{i=1}^{t} p_i} \end{aligned} \tag{1}$$

The following feature selection algorithm is the direct result of Equation 1.

Algorithm: For feature term selection, compute θ_{T_i} as follows:

$$\begin{aligned} \theta_{T_i} &= \begin{cases} \frac{f_i}{\lambda} - \frac{\alpha}{\beta} \theta_{W_i} & \text{if } 1 \leq i \leq t \\ 0 & \text{otherwise} \end{cases} \\ \lambda &= \frac{\sum_{i=1}^{t} f_i}{1 + \frac{\alpha}{\beta} \sum_{i=1}^{t} \theta_{W_i}} \end{aligned} \tag{2}$$

Then sort candidate feature terms in decreasing order of θ_{T_i}. Feature terms are those whose θ_{T_i} score satisfy a predefined confidence level. Alternatively we can simply select only the top N terms.

2.2.2. Likelihood Test. This method is based on the likelihood-ratio test by Dunning [4]. Let D_+ be a collection of documents focused on a topic T, D_- those not focused on T, and bnp a candidate feature term extracted from D_+ as defined in Section 2.1. Then, the likelihood ratio $-2log\lambda$ is defined as follows:

$$\begin{aligned} -2log\lambda &= -2log\frac{\max_{p_1 \leq p_2} L(p_1, p_2)}{\max_{p_1, p_2} L(p_1, p_2)} \\ p_1 &= p(d \in D_+ | bnp \in d) \\ p_2 &= p(d \in D_+ | \overline{bnp} \in d) \end{aligned}$$

where $L(p_1, p_2)$ is the likelihood of seeing bnp in both D_+ and D_-.

Assuming that each bnp is a Bernoulli event, the counts from Table 1 follow a binomial distribution, and the following likelihood ratio is asymptotically χ^2 distributed.

$$-2log\lambda = -2log\frac{\max_{p_1 \leq p_2} b(p_1, C_{11}, C_{11} + C_{12}) * b(p_2, C_{21}, C_{21} + C_{22})}{\max_{p_1, p_2} b(p_1, C_{11}, C_{11} + C_{12}) * b(p_2, C_{21}, C_{21} + C_{22})}$$

$$\text{where} \quad b(p, k, n) = p^k \cdot (1 - p)^{n-k}$$

$$-2log\lambda = \begin{cases} -2 * lr & \text{if } r_2 < r_1 \\ 0 & \text{if } r_2 \geq r_1 \end{cases} \tag{3}$$

$$\begin{aligned} r_1 &= \frac{C_{11}}{C_{11}+C_{12}}, \quad r_2 = \frac{C_{21}}{C_{21}+C_{22}} \\ r &= \frac{C_{11}+C_{21}}{C_{11}+C_{12}+C_{21}+C_{22}} \\ lr &= (C_{11}+C_{21})log(r) + (C_{12}+C_{22})log(1-r) - C_{11}log(r_1) \\ &\quad - C_{12}log(1-r_1) - C_{21}log(r_2) - C_{22}log(1-r_2) \end{aligned}$$

The higher the value of $-2log\lambda$, the more likely the bnp is relevant to the topic T.

	$\lvert D_+\rvert$	$\lvert D_-\rvert$	source
digital camera	485	1838	www.cnet.com www.dpreview.com www.epinions.com, www.steves-digicams.com
music	250	2389	www.epinions.com

Table 2. The product review datasets

	digital camera (38)	music (31)
BNP-M	63%	61%
dBNP-M	68%	32%
bBNP-M	32%	29%
BNP-L	68%	92%
dBNP-L	81%	96%
bBNP-L	**97%**	**100%**

Table 3. Precision of feature term extraction algorithms

Algorithm: For each bnp, compute the likelihood score, $-2log\lambda$, as defined in equation 3. Then, sort bnp in decreasing order of their likelihood score. Feature terms are all bnp's whose likelihood ratio satisfy a pre-defined confidence level. Alternatively simply only the top N bnp's can be selected.

2.3. Evaluation

2.3.1. The Dataset. We carried out experiments on two domains: digital camera and music review articles. Each dataset is a mix of manually labeled topic domain documents (D_+) and non-topic domain documents (D_-) that are randomly selected from the web pages collected by our web-crawl. The datasets are summarized in Table 2.

2.3.2. Experimental Results. We ran the two feature extraction algorithms in six different settings on the product review datasets:

- *BNP-M*: Mixture Model with *BNP*
- *dBNP-M*: Mixture Model with *dBNP*
- *bBNP-M*: Mixture Model with *bBNP*
- *BNP-L*: Likelihood Test with *BNP*
- *dBNP-L*: Likelihood Test with *dBNP*
- *bBNP-L*: Likelihood Test with *bBNP*

First, *BNP*, *dBNP*and *bBNP* were extracted from the review pages and the Mixture Model and Likelihood Test were applied on the respective *bnp's*. Terms with likelihood ratio above 0 were extracted for *bBNP-L*: 38 and 31 feature terms for digital camera and music datasets respectively. For the rest of the settings, the thresholding scheme was applied giving the same number of terms (i.e., 38 and 31 respectively for digital camera and music datasets) at the top of the lists. This thresholding gives the best possible precision scores for the other settings, since terms on the top of the list are more likely to be feature terms. We used the Ratnaparkhi POS tagger[14] to extract *bnp's*. $\alpha = 0.3$ was used for the computation of the Mixture Model. (Other values of α were used, which did not produce any better results than what are reported here.) The extracted feature terms were manually examined by two human subjects and only the terms that both subjects labeled as feature terms were counted for the computation of the precision.

Digital Camera	camera, picture, flash, lens, picture quality, battery, software, price, battery life, viewfinder, color, feature, image, menu, manual, photo, movie, resolution, quality, zoom
Music Albums	song, album, track, music, piece, band, lyrics, first movement, second movement, orchestra guitar, final movement, beat, production, chorus first track, mix, third movement, piano, work

Table 4. Top 20 feature terms extracted by *bBNP-L* **in the order of their rank**

The precision scores are summarized in Table 3. *bBNP-L* performed impressively well. The Likelihood Test method consistently performed better than the Mixture Model algorithm. Its performance continued improving with increasing level of restrictions in the candidate feature terms, perhaps because, with further restriction, the selected candidate terms are more probable feature terms. On the contrary, interestingly, the increasing level of restrictions had the reverse effect with the Mixture Model algorithm. This might be because the restrictions caused too much perturbation on term distributions for the algorithm to reliably estimate the multinomial distribution of the topic-specific model. We need further investigation to explain the behavior.

The top 20 feature terms extracted by *bBNP-L* from the digital camera and music datasets are listed in Table 4.

3. Sentiment Analysis

In this section, we describe the linguistic resources used by sentiment analysis (3.1), define the scope of sentence structures that *SA* is dealing (3.2), sentiment phrase identification and sentiment assignment (3.3), and relationship analysis (3.4).

3.1. Linguistic Resources

Sentiment about a subject is the orientation (or polarity) of the opinion on the subject that deviates from the neutral state. Sentiment that expresses a desirable state (e.g., `The picture is flawless.`) has *positive* (or "+") polarity, while one representing an undesirable state (e.g., `The product fails to meet our quality expectations.`) has *negative* (or "-") polarity. The *target* of sentiment is the subject that the sentiment is directed to: `the picture` and `the product` for the examples above. *SA* uses sentiment terms defined in the *sentiment lexicon* and sentiment patterns in the *sentiment pattern database*.

3.1.1. Sentiment Lexicon. The *sentiment lexicon* contains the sentiment definition of individual words in the following form:

```
<lexical_entry> <POS> <sent_category>
```

- `lexical_entry` is a (possibly multi-word) term that has sentimental connotation.
- `POS` is the required POS tag of `lexical entry`.
- `sentiment_category`: + | -

The following is an example of the lexicon entry:

```
    "excellent" JJ  +
```

We have collected sentiment words from several sources: General Inquirer (GI)[1], Dictionary of Affect of Language (DAL)[2][21], and WordNet[11]. From GI, we extracted all words in `Positive`, `Negative`, and `Hostile` categories. From DAL, we extracted words whose affect scores are one standard deviation higher (*positive*) or lower (*negative*) than the mean. From WordNet, we extracted synonyms of known sentiment words. At present, we have about 3000 sentiment term entries including about 2500 adjectives and less than 500 nouns.

3.1.2. Sentiment Pattern Database. Our sentiment pattern database contains sentiment extraction patterns for sentence predicates. The database entry is defined in the following form:

```
<predicate> <sent_category> <target>
```

- `predicate`: typically a verb
- `sent_category`: + | - | [˜]`source`
  `source` is a sentence component (`SP`|`OP`|`CP`|`PP`) whose sentiment is transferred to the `target`. `SP`, `OP`, `CP`, and `PP` represent subject, object, complement (or adjective), and prepositional phrases, respectively. The opposite sentiment polarity of `source` is assigned to the target, if ˜ is specified in front of `source`.
- `target` is a sentence component (`SP`| `OP`|`PP`) the sentiment is directed to.

Some verbs have positive or negative sentiment by themselves, but some verbs (we call them *trans* verb), such as "be" or "offer", do not. The sentiment of a subject in a sentence with a *trans* verb is determined by another component of the sentence. Some example sentiment patterns and sentences matching with them are:

```
impress +  PP(by;with)
      I am impressed by the picture quality.
be CP SP
      The colors are vibrant.
offer OP SP
      IBM offers high quality products.
      IBM offers mediocre services.
```

Initially, we collected sentiment verbs from GI, DAL, and WordNet. For GI and DAL, the sentiment verb extraction is the same as the sentiment term extraction as described in Section 3.1.1. From WordNet we extracted verbs from the *emotion cluster*. From the training datasets described in Section 2.3.1, we manually refined some of the patterns. The refinements typically involve the specification of sentiment source and target, as the typical error *SA* initially introduced was the association of the discovered sentiment to a wrong target. Currently, we have about 120 sentiment predicate patterns in the database.

3.2. Scope of Sentiment Analysis

As a preprocessing step to our sentiment analysis, we extract sentences from input documents containing mentions of subject terms of interest. Then, *SA* applies sentiment analysis to *kernel sentences* [7] and some text fragments. Kernel sentences usually contain only one verb. For kernel sentences, *SA* extracts the following types of *ternary expressions (T-expressions)*[7]:

- `positive` or `negative` sentiment verbs:
  `<target, verb, "">`
- `trans` verbs:
  `<target, verb, source>`

The following illustrates *T-expressions* of given sentences:

```
<the camera, like, "">
   ex. I like the camera.
<the digital zoom, be, too grainy>
   ex. The digital zoom is too grainy.
```

For text fragments, *SA* extracts *binary expressions (B-expressions)*,

```
    <adjective, target>
ex. good quality photo : <good quality, photo>
```

3.3. Sentiment Phrases and Sentiment Assignment

After parsing each input sentence by a syntactic parser, *SA* identifies sentiment phrases from subject, object, adjective, and prepositional phrases of the sentence.

Adjective phrases: Within the phrase, we identify all sentiment adjectives defined in the sentiment lexicon. For example, `vibrant` is positive sentiment phrase for the sentence "`The colors are vibrant.`"

Subject, object and prepositional phrases: We extract all base noun phrases of the forms defined in Section 2.1.1 that consist of at least one sentiment word. The sentiment of the phrase is determined by the sentiment words in the phrase. For example, `excellent pictures` (`JJ NN`) is a positive sentiment phrase because `excellent` (`JJ`) is a positive sentiment word. For a sentiment phrase with a word with negative meaning, such as `not`,`no`, `never`, `hardly`, `seldom`, or `little`, the polarity of the sentiment is reversed.

[1] http://www.wjh.harvard.edu/∼inquirer/
[2] http://www.hdcus.com

3.4. Semantic Relationship Analysis

SA extracts *T*- and *B-expressions* in order to make (subject, sentiment) association. From a *T-expression*, sentiment of the *verb* (for sentiment verbs) or *source* (for *trans* verb), and from a *B-expression*, sentiment of the *adjective*, is assigned to the *target*.

3.4.1. Sentiment Pattern based Analysis. For each sentiment phrase detected (Section 3.3), *SA* determines its *target* and final polarity based on the sentiment pattern database (Section 3.1.2). *SA* first identifies the *T-expression*, and tries to find matching sentiment patterns. Once a matching sentiment pattern is found, the *target* and sentiment assignment are determined as defined in the sentiment pattern.

Some sentiment patterns define the *target* and its sentiment explicitly. Suppose the following sentence, sentiment pattern, and subject is given:

```
I am impressed by the flash capabilities.
pattern : "impress" + PP(by;with)
subject : flash
```

SA first identifies the *T-expression* of the sentence: `<flash capability, impress, "">` and directly infers that the *target* (`PP` lead by "by" or "with"), `the flash capabilities`, has positive sentiment: `(flash capability, +)`.

For sentences with a *trans* verb, *SA* first determines the sentiment of *source*, and assigns the sentiment to the *target*. For example, for the following sentence and the given subject term `camera`:

```
This camera takes excellent pictures.
```

SA first parses the sentence and identifies:

- matching sentiment pattern: `<"take" OP SP>`
- subject phrase (`SP`): `this camera`
- object phrase (`OP`) : `excellent pictures`
- sentiment of the `OP`: `positive`
- *T-expression* : `<camera, take, excellent picture>`

From this information, *SA* infers that the sentiment of *source* (`OP`) is positive, and associates positive sentiment to the *target* (`SP`): `(camera, +)`.

During the semantic relationship analysis, *SA* takes *negation* into account at the sentence level: if an adverb with negative meaning (such as `not`, `never`, `hardly`, `seldom`, or `little`) appears in a verb phrase, *SA* reverses the sentiment of the sentence assigned by the corresponding sentiment pattern. For example, *SA* detects negative polarity from the following sentence:

```
This camera is not good for novice users.
```

3.4.2. Analysis without Sentiment Pattern. There are many cases where sentiment pattern based analysis is not possible. Common cases include:

- No corresponding sentiment pattern is available.
- The sentence is not complete.
- Parser failure, possibly due to missing punctuation, wrong spelling, etc.

	Precision	Recall	Accuracy
SA	87%	56%	85.6%
Collocation	18%	70%	N/A
ReviewSeer	N/A	N/A	88.4%

Table 5. Performance comparison of sentiment extraction alorithms on the product review datasets.

Examples of fragments containing sentiment are:

```
Poor performance in a dark room. (1)
Many functionalities for the price. (2)
```

SA creates *B-expressions* and makes the sentiment assignment on the basis of the phrase sentiment. The *B-expressions* and sentiment associations of sentences (1) and (2) are:

(1^B) `<poor, performance>   : (performance, -)`
(2^B) `<many, functionality> : (functionaliry, +)`

3.5. Evaluation

For experiments, we used the Talent[3] shallow parser for sentence parsing, and *bBNP-L* for feature extraction.

3.5.1. Product Review Dataset. We ran *SA* on the review article datasets (Section 2.3.1). The review articles are a special class of web documents that typically have a high percentage of sentiment-bearing sentences. For each subject term, we manually assigned the sentiment. Then, we ran *SA* for each sentence with a subject term and compared the computed sentiment label with the manual label to compute the accuracy. The result is compared with the collocation algorithm and the best performing algorithm of *ReviewSeer*[3]. To our knowledge, *ReviewSeer* is by far the latest and the best opinion classifier. The collocation algorithm assigns the polarity of a sentiment term to a subject term, if the sentiment term and the subject term exist in the same sentence. If positive and negative sentiment terms coexist, the polarity with more counts is selected.

The overall precision and recall of *SA* are 87% and 56%, respectively (Table 5). The accuracy of the best performing algorithm of *ReviewSeer* is 88.4% (vs. 85.6% of *SA*). The precision of the Collocation algorithm is significantly lower, only 18%, as expected, with high recall of 70%.

Although the results provide a rough comparison, they are not directly comparable. First, the test datasets are not the same. Although both *SA* and *ReviewSeer* use product review articles, the actual datasets are not identical. (We are not aware of any benchmark dataset for sentiment classification for evaluation purposes.) They have combined more categories (7 categories vs. 2 categories for *SA*). Secondly,

[3] http://flahdo.watson.ibm.com/Talent/talent_project.htm

	Precision	Accuracy	Acc. w/o *I class*
SA(Petroleum, Web)	86%	90%	N/A
SA(Pharmaceutical, Web)	91%	93%	N/A
SA(Petroleum, News)	88%	91%	N/A
$ReviewSeer$ (Web)	N/A	38%	68%

Table 6. The performance of *SA* and *ReviewSeer* on general web documents and news articles.

ReviewSeer is a document level sentiment classifier, while *SA* is per subject-spot level. Third, *ReviewSeer* does not try to do subject association.

ReviewSeer might have produced better accuracy with fewer categories. On the other hand, since they do not try (subject, sentiment) association, their accuracy is not affected by the potential association error, while *SA*'s is. That is, even though *SA* extracts sentiment polarity accurately, we consider it a failure if the (subject, sentiment) association is made wrong. It is not clear how much the subject association would impact *ReviewSeer*'s accuracy. However, the experimental results on general web documents (Section 3.5.2) reveal how much subject association error degrades, at least partially, the accuracy of *ReviewSeer*.

3.5.2. General Web Documents. Sentiment expressions in general Web documents are typically very sparse in comparison to the review articles. This characteristic of general web documents may work against a document level classifier as there might not be enough sentiment-bearing expressions in a document to classify the entire document as sentiment-bearing.

In order to mitigate the problem, *ReviewSeer* applied the algorithm on the individual sentences with a subject word. This makes the comparison with *SA* on more equal ground. Table 6 lists the results.

SA achieves high precision (86% $\sim$ 91%) and even higher accuracy (90% $\sim$ 93%) on general Web documents and news articles. The precision of *SA* was computed only on the test cases that *SA* extracted as either $positive$ or $negative$, but did not include $neutral$ cases. The accuracy of *SA* included the $neutral$ cases as well, as did $ReviewSeer$'s. The accuracy of *SA* is higher than the precision, because the majority of the test cases do not have any sentiment expression, and *SA* correctly classifies most of them as neutral.

On the contrary, *ReviewSeer* suffered with sentences from general web documents: the accuracy is only 38% (down from 88.4%). (The accuracy is computed based on the figures from Table 14 of [3]: we have averaged the accuracies of the three equal-size groups of a test set, 21%, 42% & 50%, respectively.) The accuracy was improved to 68% after removing difficult cases and using only clearly positive or negative sentences about the given subject. The set of difficult testing cases eliminated (called *I class*) include sentences that were ambiguous when taken out of context (*case i*), were not describing the product (*case ii*), or did not express any sentiment at all (*case iii*).

The challenge here is that these difficult cases are the majority of the sentences that any sentiment classifier has to deal with: 60% (356 out of 600) of the test cases for the *ReviewSeer* experiment and even more (as high as over 90% on some domain) in our experiments. *Case i* is difficult for any sentiment classifier. We believe *case ii* is where the purely statistical methods do not perform well and sophisticated NLP can help. *SA* tries to solve the (subject, sentiment) association problem *case ii* by the relationship analysis. *SA* handles the neutral cases *iii* already very well as discussed earlier.

4. Previous Work

[1] describes a procedure that aims at extracting *part-of* features, using possessive constructions and prepositional phrases, from news corpus. By contrast, we extract both *part-of* and *attribute-of* relations.

Some of the previous works on sentiment-based classification focused on classifying the semantic orientation of individual words or phrases, using linguistic heuristics, a preselected set of seed words, or by human labeling [5, 21]. [5] developed an algorithm for automatically recognizing the semantic orientation of adjectives. [22] identifies *subjective* adjectives (or sentiment adjectives) from corpora.

Past work on sentiment-based categorization of entire documents has often involved either the use of models inspired by cognitive linguistics [6, 16] or the manual or semi-manual construction of discriminant-word lexicons [2, 19]. [6] proposed a sentence interpretation model that attempts to answer directional queries based on the deep argumentative structure of the document, but with no implementation detail or any experimental results. [13] compares three machine learning methods (Naive Bayes, maximum entropy classification, and SVM) for sentiment classification task. [20] used the average “semantic orientation” of the phrases in the review. [15] analysed emotional affect of various corpora computed as average of affect scores of individual affect terms in the articles. The sentiment classifiers often assumes 1) each document has only one subject, and 2) the subject of each document is known. However, these assumptions are often not true, especially for web documents. Moreover, even if the assumptions are met, sentiment classifiers are unable to reveal the sentiment about individual features, unlike *SA*.

Product Reputation Miner [12] extracts positive or negative opinions based on a dictionary. Then it extracts characteristic words, co-occurrence words, and typical sentences for individual target categories. For each characteristic word or phrase they compute frequently co-occurring terms. However, their association of characteristic terms and co-occurring terms does not necessarily mean relevant opinion as was seen in collocation experiments. In contrast, our NLP based relationship analysis associates subjects to the corresponding sentiments.

ReviewSeer [3] is a document level opinion classifier that uses mainly statistical techniques and some POS tagging information for some of their text term selection algorithms. It achieved high accuracy on review articles. However, the performance sharply degrades when applied to sentences with subject terms from the general web documents. In contrast, *SA* continued to perform with high accuracy. Unlike *ReviewSeer*, *SA* handles the neutral cases and subject association very well. In fact, the relationship analysis of *SA* was designed for these kinds of difficult cases.

5. Discussion and Future Work

We applied NLP techniques to sentiment analysis. The feature extraction algorithm successfully identified topic related feature terms from online review articles, enabling sentiment analysis at finer granularity. *SA* consistently demonstrated high quality results of 87% for review articles, 86 $\sim$ 91% (precision) and 91 $\sim$ 93% (accuracy) for the general web pages and news articles. The results on review articles are comparable with the state of the art sentiment classifiers, and the results on general web pages are better than those of the state of the art algorithms by a wide margin (38% vs. 91 $\sim$ 93%).

However, from our initial experience with sentiment detection, we have identified a few areas of potentially substantial improvements. We expect full parsing will provide better sentence structure analysis, thus better relationship analysis. Second, more advanced sentiment patterns currently require a fair amount of manual validation. Although some amount of human expert involvement may be inevitable in the validation to handle the semantics accurately, we plan on more research on increasing the level of automation.

References

[1] M. Berland and E. Charniak. Finding parts in very large corpora. In *Proc. of the 37th ACL Conf.*, pages 57–64, 1999.

[2] S. Das and M. Chen. Yahoo! for anazon: Extracting market sentiment from stock message boards. In *Proc. of the 8th APFA*, 2001.

[3] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proc. of the 12th Int. WWW Conf.*, 2003.

[4] T. E. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1), 1993.

[5] V. Hatzivassiloglou and K. R. McKeown. Predicting the semantic orientation of adjectives. In *Proc. of the 35th ACL Conf.*, pages 174–181, 1997.

[6] M. Hearst. Direction-based text interpretation as an information access refinement. *Text-Based Intelligent Systems*, 1992.

[7] B. Katz. From sentence processing to information access on the world wide web. In *Proc. of AAAI Spring Symp. on NLP*, 1997.

[8] H. Li and K. Yamanishi. Mining from open answers in questionnaire data. In *Proc. of the 7th ACM SIGKDD Conf.*, 2001.

[9] C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[10] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19, 1993.

[11] G. A. Miller. Nouns in WordNet : A lexical inheritance system. *Int. J. of Lexicography*, 2(4):245–264, 1990. Also available from ftp://ftp.cogsci.princeton.edu/pub/wordnet/5papers.ps.

[12] S. Morinaga, K. Yamanishi, K. Teteishi, and T. Fukushima. Mining product reputations on the web. In *Proc. of the 8th ACM SIGKDD Conf.*, 2002.

[13] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proc. of the 2002 ACL EMNLP Conf.*, pages 79–86, 2002.

[14] A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proc. of the EMNLP Conf.*, pages 133–142, 1996.

[15] L. Rovinelli and C. Whissell. Emotion and style in 30-second television advertisements targeted at men, women, boys, and girls. *Perceptual and Motor Skills*, 86:1048–1050, 1998.

[16] W. Sack. On the computation of point of view. In *Proc. of the 12th AAAI Conf.*, 1994.

[17] P. Subasic and A. Huettner. Affect analysis of text using fuzzy semantic typing. *IEEE Trans. on Fuzzy Systems, Special Issue*, Aug., 2001.

[18] L. Terveen, W. Hill, B. Amento, D. McDonald, and J. Creter. PHOAKS: A system for sharing recommendations. *CACM*, 40(3):59–62, 1997.

[19] R. M. Tong. An operational system for detecting and tracking opinions in on-line discussion. In *SIGIR Workshop on Operational Text Classification*, 2001.

[20] P. D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proc. of the 40th ACL Conf.*, pages 417–424, 2002.

[21] C. Whissell. The dictionary of affect in language. *Emotion: Theory, Research, and Experience*, pages 113–131.

[22] J. M. Wiebe. Learning subjective adjectives from corpora. In *Proc. of the 17th AAAI Conf.*, 2000.

[23] C. Zhai and J. Lafferty. Model-based feedback in the lanuage modeling approach to information retrieval. In *Proc. of the 10th Information and Knowledge Management Conf.*, 2001.

[24] Y. Zhang, W. Xu, and J. Callan. Exact maximum likelihood estimation for word mixtures. In *ICML Workshop on Text Learning*, 2002.