Sentiment Mining in WebFountain

Jeonghee Yi Wayne Niblack

IBM Almaden Research Center
650 Harry Rd.
San Jose, CA 95120

Abstract

WebFountain is a platform for very large-scale text analytics applications that allows uniform access to a wide variety of sources. It enables the deployment of a variety of document-level and corpus-level miners in a scalable manner, and feeds information that drives end-user applications through a set of hosted Web services.

Sentiment (or opinion) mining is one of the most useful analyses for various end-user applications, such as reputation management. Instead of classifying the sentiment of an entire document about a subject, our sentiment miner determines sentiment of each subject reference using natural language processing techniques. In this paper, we describe the fully functional system environment and the algorithms, and report the performance of the sentiment miner. The performance of the algorithms was verified on online product review articles, and more general documents including Web pages and news articles.

1 Introduction

WebFountain [8] is a platform for very large-scale text analytics applications that allows uniform access to a wide variety of sources, including large portions of the Web, traditional news feeds, bulletin boards, analyst reports, and both structured and unstructured application-specific data. It enables scalable deployment of document- and corpus-level *miners* (text analytics softwares) that analyze the data and produce information that drives end-user applications through a set of hosted Web services. Tokenization, geographic context extraction[15], template detection[3], and page ranking[27] are examples of tasks the miners perform. Users of an application system are able to remotely develop analytical components using a collection of Web service APIs (Application Programming Interfaces). The system is fully operational: it crawls, stores, processes, and analyzes billions of documents and hundreds of terabytes of information [8].

For a proof of concept, a reputation management application has been built on the WebFountain platform that enables various analyses for corporate customers, including analysis on their corporate and product reputation, and tracking of market trends.

A key component of our reputation management system is the *sentiment miner* that extracts sentiment (or opinions) people express about a *subject*, such as a company, brand, or product name. Knowing the reputation of their own or their competitors' products or brands is highly valuable for product development, marketing and consumer relationship management. The WebFountain platform offers a rich and diverse set of data that, if properly analyzed, can replace (or at least complement) costly and time-consuming consumer surveys.

Opinions in natural language are often expressed in subtle and complex ways, presenting challenges which are not easily addressed by standard text categorization approaches. Negative reviews may contain many apparently positive phrases even while maintaining a strong negative tone, and the opposite is also common. Although effective applications of standard text mining algorithms for sentiment analysis on a limited domain have been reported [21, 6, 19], there still exists significant gap between the quality the existing solutions offer and the requirements corporate customers have in order to be able to adopt the solutions in making their business decisions. Moving beyond the current approaches may involve addressing the task at several levels, from recognizing the semantic impact of individual words or phrases to a complex discourse analysis.

In the rest of this section, we briefly review related work in sentiment analysis, and then describe our design goals in building the sentiment miner.

1.1 Related Works

There has been extensive research on automatic text analysis for sentiment, such as sentiment classifiers[4, 5, 10, 18, 20, 21, 24, 28], affect analysis[25, 30], automatic survey analysis[12, 24], opinion extraction[17], and recommender systems [26]. These methods typically try to extract the overall sentiment revealed in a document, either positive or negative, or somewhere in between.

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 [9, 30]. [9] developed an algorithm for automatically recognizing the semantic orientation of adjectives. [31] identifies *subjective*

Proceedings of the 21st International Conference on Data Engineering (ICDE 2005)
1084-4627/05 $20.00 © 2005 **IEEE**

adjectives (or sentiment adjectives) from corpora.

Most of the past work on sentiment-based categorization assumes that an entire document is only about a subject, and apply (a variation of) existing text classification algorithms. They often involve either the use of models inspired by cognitive linguistics [10, 24] or the manual or semi-manual construction of discriminant-word lexicons [5, 28, 11]. [10] 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. [21] compares three machine learning methods (Naive Bayes, maximum entropy classification, and SVM) for sentiment classification task. [20] applies text categorization only on the subjective portions of the document. [29] used the average "semantic orientation" of the phrases in the review. [23] analyzed emotional affect of various corpora computed as average of affect scores of individual affect terms in the articles. [4] extended a sentiment classification model that utilize unlabeled documents and human-provided information as well as labeled documents. Most of the sentiment classifiers assume 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.

Product Reputation Miner [17] 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 collocation-based association of characteristic terms and co-occurring terms is known to be highly noisy [6].

ReviewSeer [6] 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, but the performance sharply degrades when applied to sentences with subject terms from the general web documents.

1.2 The Sentiment Miner

Our sentiment miner is designed with the following two challenges in mind: First, not only the overall opinion about a topic, but also sentiment about individual aspects of the topic is essential information of interest. For example, though one is generally happy about a digital camera, he might be dissatisfied by the short battery life. To the manufacturers, these individual weaknesses and strengths are important to know, perhaps more valuable than the overall satisfaction level of customers. Document level sentiment classification fails to detect sentiment about individual aspects of the topic.

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 [6]. 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 research 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*[6] 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.`
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.`

ReviewSeer 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, which is wrong for `T series CLIEs`. Also 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 designed and developed our sentiment miner to mitigate these shortcomings and to produce the following output for the sample sentences above provided that `Sony PDA`, `NR70`, and `T series CLIEs` are pre-specified topics:

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

Our sentiment miner analyzes grammatical sentence structures and phrases based on natural language processing (NLP) techniques. It detects, for each occurrence of a known topic spot, the sentiment specifically about the topic. It consists of the following components:

- subject spotting
- topic-specific feature extraction (optional)
- sentiment extraction for each sentiment-bearing phrase
- sentiment assignment to an appropriate topic

The rest of this paper is organized as follows: In Section 2, we describe the WebFountain platform system environment that the sentiment miner runs on. Section 3 provides an overview of the sentiment mining system. Section **??** describes the feature term extraction algorithm and the sentiment detection algorithms, and reports the experimental results. Finally, we conclude with a discussion in Section 5.

2 The WebFountain System Overview

Figure 1 shows an overview of the WebFountain system. For more detailed description of the system, refer to [8]. The system is designed as a loosely coupled, shared-nothing parallel cluster of Intel-based Linux servers. Currently there are more than 500 machines – IBM xSeries servers (model x335 and x350) with 2.4 GHz Intel Xeon processors with 4 gigabits RAM, and 2560 72 GB hard disk drives arranged into 512 five-disk RAID5 arrays for data storage – managed by the cluster manager. The nodes in the cluster communicate using a Web-service style, lightweight, high-speed communication protocol called *Vinci* [1], a derivative of Simple Object Access Protocol (SOAP). The WebFountain system achieves scalability of up to billions of documents by full parallelism.

There are two processes for a highly parallel data loading into WebFountain. Large-scale Web content acquisition is done by Web crawlers. Acquisition of other sources, such as traditional news feeds, preprocessed bulletin boards, NNTP, and a variety of both structured and unstructured customer data is done by a set of *ingestors* that handle the unique delivery method and format of each source. Each data source comes with its own unique delivery method and formatting.

The WebFountain *data store* component manages entities that are represented in XML (Extensible Markup Language). An entity is a referenceable unit of information such as a Web page. The data store stores, modifies, and retrieves entities. The WebFountain *indexer* creates indices not only from text tokens from raw entities, but also from conceptual tokens generated by miners. The indexer supports multiple indices for various query types including boolean, range, regular expression, spherical, and other complex query types.

There are two types of miners in WebFountain: entity-level and corpus-level (cross-entity) miners. Entity-level miners process each entity without information from neighboring entities, and typically augment processed entities with the results. Examples of entity-level miners include tokenizers, geographic context discoverer, named entity extractor, and machine translator. In contrast, corpus-level miners require all or part of the entire data in store in order to perform their tasks. Examples of entity-level miners are computing aggregate statistics, duplicate detection, trending, and clustering.

3 The Sentiment Mining System

The sentiment miner is an entity-level miner that has two operational modes. The first is for applications in which a set of subjects of interest is available. The second is for applications where no subject of interest is provided a priori but is given at query time.

Sentiment mining with a predefined set of subjects: Some application users have a predefined list of subjects they are interested in analyzing comprehensively. In one common application, end users know a set of subjects, products, and brands they want to track. Figure 2 depicts an overview of the sentiment mining process for such cases.

Given the set of subjects, the documents are preprocessed to identify the occurrences of the subjects (by the *spotter*). The spots are filtered by the *disambiguator* to identify spots of a given subject term that are related to the intended topic. Feature terms of the subject terms can be given by the end-users or automatically identified by the feature extractor[32]. A small sentiment context for each subject term spot is constructed and the sentiment miner runs on the context. A sentiment context generally consists of the full sentence that contains a subject spot and possibly some surrounding text of the sentence determined by the sentiment context window formation rule. The subject spot is marked by an XML tag and passed to the sentiment analyzer. The sentiment analyzer uses the information in the *predicate rule database* and the *sentiment term dictionary* to perform the analysis. The sentiments identified are stored in a database to be fed into user applications.

Sentiment mining with no predefined set of subjects: There are application environments where the list of interesting subjects is not predefined and users need to pose sentiment queries about any random subjects. Once the query is given, the system could, in principle, search for the subject terms, identify subject spots (by using index), build corresponding sentiment contexts, and apply the sentiment analysis at run time. This runtime execution of sentiment analysis is too slow for most users expecting real time response: the size of the corpus WebFountain applications typically run on is too big for real time processing, especially for computationally intensive miners such as the sentiment miner.

An alternative approach is to apply sentiment analysis to the entire corpus offline and index on them for quick query time response. Figure 3 illustrates the sentiment mining in this mode.

In order to identify potential subject terms, documents are preprocessed to identify named entities. We use a simple named entity spotter that detects all capitalized nouns

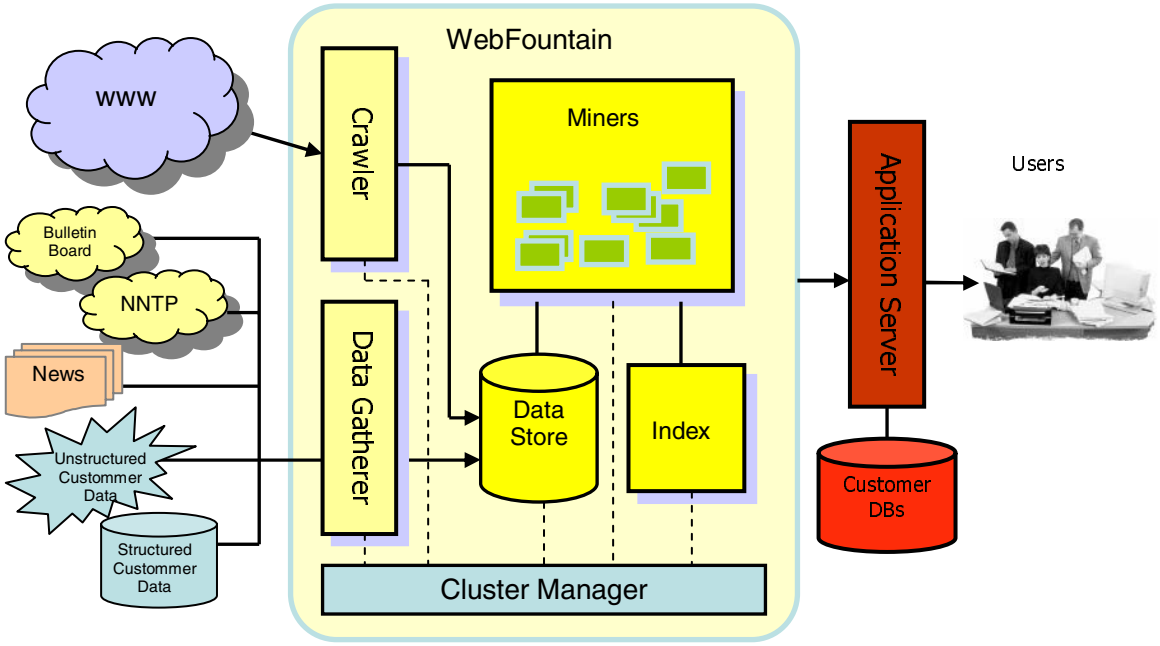

Figure 1. The WebFountain System Architecture

(by the *named entity spotter*) as subjects, and extract a corresponding sentiment context. A sentiment context is created for each sentence with a name entity and the named entity is spotted as a subject term. In addition, the sentiment miner spots sentiment terms and analyzes each sentiment-bearing sentence. The sentiment analyzer is configured to identify a corresponding phrase (such as a subject or object phrase of the sentence) that associates with a sentiment term. The sentiment and the entity (or subject) associated with the sentiment are indexed. The index is used to serve real time user queries.

In the following, we provide short descriptions of four miners that the sentiment miner has dependency on.

The Tokenizer: The tokenizer produces a stream of tokens from the input text. This is language dependent, and thus the actual tokenizer used from each entity is determined by the input text language.

The Spotter: The spotter is a general purpose miner that identifies occurrences of arbitrary terms or phrases within documents. The spotter is given a list of terms to identify and it tags documents that contain them with tokens specifying where the terms appear. Subject terms are grouped into synonym sets that are user configurable and the spotter annotates the occurrences with the synonym set ID. This enables us to count the different variations on a subject name all together when doing analytics on occurrences of a subject. Subject occurrences identified by the spotter are referred to as *spots*.

The Disambiguator: The disambiguator [2] evaluates each spot to determine if it is truly related to the intended subject. For instance, suppose the given subject term is "SUN Microsystems." The disambiguator determines if an occurrence of text token "SUN" refers to the subject (on topic), or something else like "Sunday" (off topic).

Disambiguation is an essential process for identifying mentions of a given subject term because, due to the high ambiguity of natural language, some token strings that match the subject term may not refer to the intended subject, as shown in the example. Our disambiguation system is based on the classical idea that disambiguation can be achieved by relying on the presence or absence of additional terms that appear in the context of a subject. It utilizes user-defined sets of terms that are *positively* (or *negatively*) related to the topic for each domain. For each spot, it computes a score for a local context surrounding the spot, and a global context (the full document). The score is based on the on-topic and off-topic terms formed, their $TF * IDF$ scores, and their types (single term or lexical affinity [2]). If the global context score passes a threshold, all spots on the page are considered on-topic. Otherwise it checks whether the combined local context and global context score passes another threshold to determine whether the particular spot is on-topic.

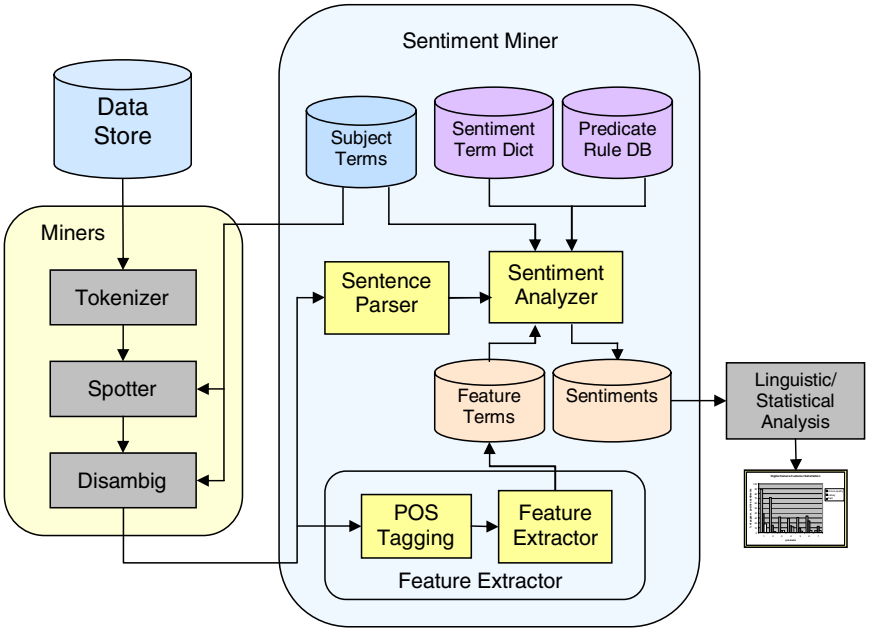

Figure 2. The Sentiment Mining Process with a Predefined Set of Subjects

The Named Entity Spotter: The named entity spotter detects all capitalized noun phrases. From the tokenized text documents, it forms a candidate name list by collecting sequences of capitalized tokens as well as special lowercase tokens such as 'and' and 'of'. Each candidate name is examined for the presence of conjunctions, prepositions or possessives. These may indicate the candidate name has to be split into multiple components. A set of heuristics is applied to each candidate name to determine where the split has to be made. For example, the candidate name *Prof. Wilson of American University* is split into two different named entities *Prof. Wilson* and *American University*.

4 The Sentiment Miner

4.1 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.

For instance, for the digital camera domain, a feature can be a *part of* the camera, such as `lenses`, `battery` or `memory card`; an *attribute* of the camera, such as `price` or `size`; or an *attribute of a feature*, such as `battery life` (an attribute of feature `battery`).

Using a set of feature term selection heuristics, we first extract noun phrases from a collection of documents focused on a certain topic, such as product reviews. The feature term selection algorithm is applied to the candidate noun phrases.

We developed a set of feature term extraction heuristics and selection algorithms [32]. The best performing candidate feature term extraction heuristic and the feature term selection algorithm combination was the likelihood ratio [7] test on terms extracted with the *bBNP* heuristic described below.

The Candidate Feature Term Extraction: *bBNP* (Beginning definite Base Noun Phrases) heuristic extracts definite base noun phrases at the beginning of sentences followed by a verb phrase. A definite base noun phrase is a noun phrase of the following patterns preceded by the definite article 'the':

- `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[14].

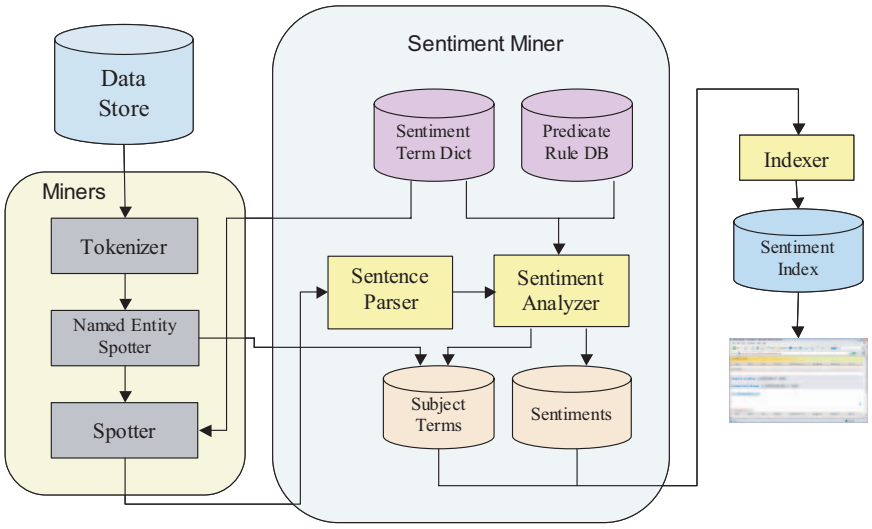

Figure 3. The Sentiment Mining Process without a Predefined Set of Subjects

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

Table 1. Counts for a base noun phrase (*bnp*) [13]

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. In addition, 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.

Feature Term Selection Algorithms: The following algorithm is based on the likelihood ratio [7] test by Dunning:

- For each bnp (base noun phrase), compute the likelihood score, $-2log\lambda$:

$$-2log\lambda = \begin{cases} -2 * lr & \text{if } r_2 < r_1 \\ 0 & \text{if } r_2 \geq r_1 \end{cases} \quad (1)$$

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

$$\begin{aligned} r_1 &= \frac{C_{11}}{C_{11}+C_{12}} \\ r_2 &= \frac{C_{21}}{C_{21}+C_{22}} \\ r &= \frac{C_{11}+C_{21}}{C_{11}+C_{12}+C_{21}+C_{22}} \end{aligned}$$

where C_{11} and C_{12} are the number of documents containing a candidate feature term, bnp, in D_+ and D_-, respectively (See Table 1). D_+ is a collection of documents focused on a topic, and D_- those not focused on the topic. Likewise, C_{21} and C_{22} represent the number of documents not containing the bnp in D_+ and D_-, respectively.

- Sort the bnp in decreasing order of their likelihood score.
- Feature terms are all bnp's whose likelihood ratio surpass a pre-defined confidence level. Alternatively select the top N bnp's.

Assuming that each bnp is a Bernoulli event, the counts C_{ij}, from Table 1 follow a binomial distribution, the likelihood ratio $-2log\lambda$ is asymptotically χ^2 distributed. The higher the likelihood ratio, the more likely the bnp is relevant to the topic T.

The Product Review Dataset: We tested the algorithms on two domains: digital cameras and music reviews. Each dataset is a mix of manually labeled topic domain documents (D_+) and non-topic domain documents (D_-) that are randomly selected from web pages collected by our web-crawl. For digital camera domain, there are 485 documents for D_+ and 1838 documents for D_- collected from the following sites:

	Digital Camera	Music Albums
1	camera	song
2	picture	album
3	flash	track
4	lens	music
5	picture quality	piece
6	battery	band
7	software	lyrics
8	price	first movement
9	battery life	second movement
10	viewfinder	orchestra
11	color	guitar
12	feature	final movement
13	image	beat
14	menu	production
15	manual	chorus
16	photo	first track
17	movie	mix
18	resolution	third movement
19	quality	piano
20	zoom	work

Table 2. Top 20 feature terms extracted by *bBNP-L* in the order of their rank

```
www.cnet.com
www.dpreview.com
www.epinions.com
www.steves-digicams.com
```

For music domain, there are 250 and 2389 documents for D_+ and D_-, respectively, collected from `www.epinions.com`.

The Experimental Results: First, base noun phrases were extracted using the *bBNP* heuristic from the review pages and the likelihood ratio test was applied. The Ratnaparkhi POS tagger[22] was used to extract *bnp's*. 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.

The feature extraction algorithm achieved very high precision: 97% and 100% for the digital camera and music domains, respectively. The top 20 feature terms extracted by the combination of the algorithms from the digital camera and music datasets are listed in Table 2. The occurrences of the selected feature terms in D_+ of digital camera dataset were counted in Table 3. There were 13 proucts reviewed in the dataset. 55 Feature terms were identified from the dataset. Feature terms were significantly more frequently referenced (13 times more) than the product names; a rough indicator of the frequency of sentiment expressions involving the feature terms.

Product Names	# of references	Feature Terms	# of references
Canon	829	camera	6554
Nikon	411	picture	2564
Sony	346	feature	2206
Olympus	297	quality	2112
Kodak	256	battery	1770
Fuji	164	zoom	1034
Minolta	106	flash	912
15 Products	2474	55 Features	30616

Table 3. Product name vs. feature term references in the digital camera dataset

4.2 Sentiment Analysis

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. Opinions have a *source* and a *target*. A *source* may be the writer or the third person mentioned in the text. The *target* of sentiment is the subject that the sentiment is directed to: `the picture` and `the product` for the examples above.

As a preprocessing step to our sentiment analysis, we extract sentences from input documents containing mentions of subject terms of interest, and sentiment terms, if necessary. The sentences are parsed by a language specific syntactic parser. The sentiment miner determines the sentiment of each phrase of a sentence parse and assigns a sentiment to a subject based on the relationship analysis. There are two major linguistic resources used for sentiment analysis: the sentiment lexicon and the sentiment pattern data base. The former defines the sentiment polarities of terms. The latter contains the sentiment assignment patterns of predicates.

The 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 the `lexical entry`.
- `sent_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][30], and WordNet[16]. The terms and their polarity were manually validated after the extraction. At present, we have about 3000 sentiment term entries including about 2500 adjectives and less than 500 nouns.

The Sentiment Pattern Database: The database entry defines a sentiment extraction pattern for a sentence predicate in the following form:

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

- `predicate`: 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 matching sentences are:

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

Sentiment Pattern Matching and Sentiment Assignment Sentiment polarity is assigned to a subject by semantic relationship analysis. After parsing each input sentence by a syntactic parser[3], the sentiment miner identifies the predicate of the sentence from the parse and searches the sentiment pattern database to find the best matching sentiment pattern of the predicate. Then, the *target* and sentiment assignment are by relationship analysis based on the information in the sentiment pattern.

Some sentiment patterns define the *target* and its sentiment explicitly. The sentiment miner assigns the sentiment defined in the matching predicate pattern to the target. For the following example sentence,

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

The sentiment miner assigns positive sentiment to the subject: `(flash capability, +)`.

For sentences with a *trans* verb, the sentiment miner first determines the sentiment of the *source* phrase, and assigns the sentiment to the *target*. The sentiment of a 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 an adverb with negative meaning, such as `not`, `no`, `never`, `hardly`, `seldom`, or `little` the sentiment polarity of the phrase is reversed.

Suppose the given subject term is `camera` for the following sentence:

```
This camera takes excellent pictures.
  - predicate: take
  - pattern: <"take" OP SP>
  - subject phrase (SP): this camera
  - object phrase (OP) : excellent pictures
  - sentiment of the OP: positive
```

The sentiment miner infers the sentiment of *source* (`OP`) is positive, and associates positive sentiment to the *target* (`SP`): `(camera, +)`.

During the semantic relationship analysis, the sentiment miner takes *negation* into account at the sentence level: if an adverb with negative meaning appears in a verb phrase, the sentiment miner reverses the sentiment of the sentence assigned by the corresponding sentiment pattern.

Evaluation on the Product Review Dataset We ran the sentiment miner on the two product review datasets described in Section 4.1. 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 the sentiment miner 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*[6]. We compare with *ReviewSeer* because 1) it is one of the best performing sentiment classification algorithms, and 2) they have evaluated their performance on

[1] http://www.wjh.harvard.edu/~inquirer/

[2] http://www.hdcus.com

[3] We used the Talent shallow parser for syntactic parsing: http://flahdo.watson.ibm.com/Talent/talent_project.htm

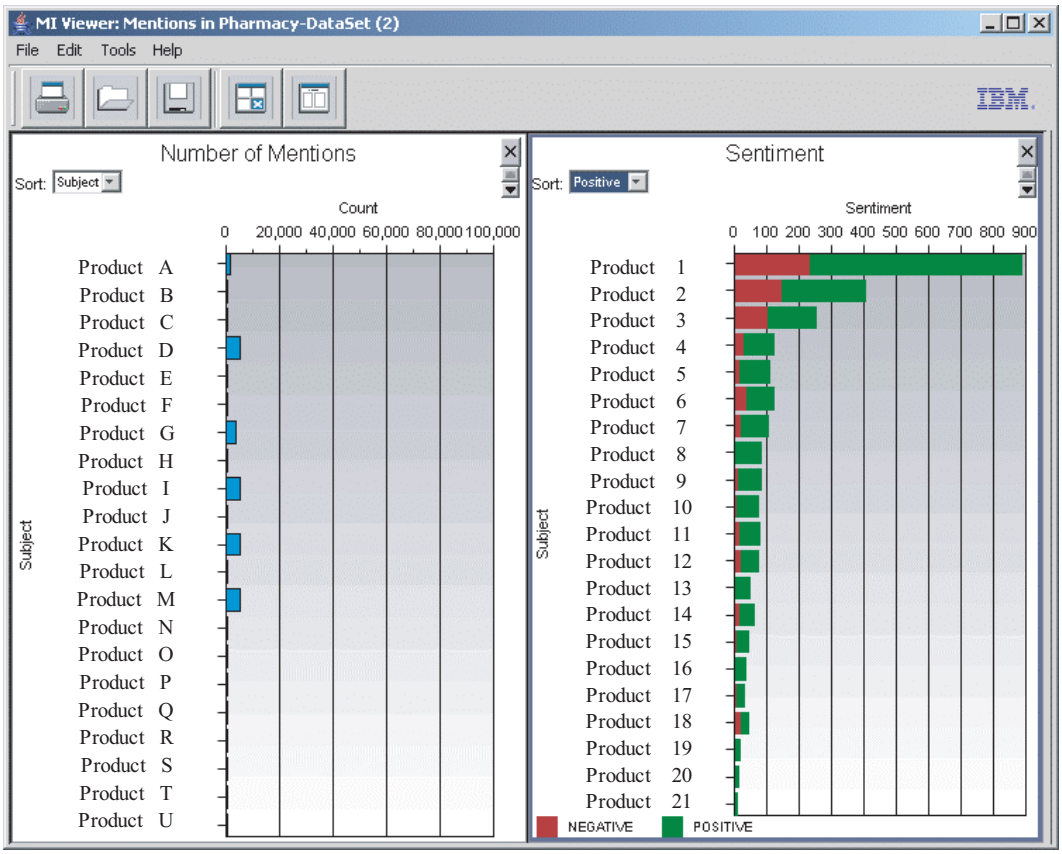

Figure 4. WebFountain GUI Visualization of the Sentiment Mining Result.

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

Table 4. Performance comparison of sentiment extraction algorithms on the product review datasets.

both product review datasets and general Web documents. The collocation algorithm assigns the polarity of a sentiment term to a subject term in the same sentence. If positive and negative sentiment terms co-exist, the polarity with more counts is selected.

The overall precision and recall of the sentiment miner are 87% and 56%, respectively (See Table 4). The accuracy of the best performing algorithm of *ReviewSeer* is 88.4% (vs. 85.6% of *SM*). The precision was computed only on the test cases with either *positive* or *negative* sentiment. For the computation of the accuracy, *neutral* sentiment cases were included as well, as *ReviewSeer* did. The sentiment miner's accuracy is higher than the precision, because the majority of the test cases have neutral sentiment, and it correctly classifies them. The precision of the Collocation algorithm is significantly lower, only 18%, as expected, with high recall of 70%.

Evaluation on General Web Documents Document level classifiers do not work as well on general Web pages in which sentiment expressions are typically very sparse, 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. Table 5 lists the results.

The sentiment miner consistently achieves high precision (86% $\sim$ 91%) and even higher accuracy (90% $\sim$ 93%) on general Web documents and news articles. On the contrary, *ReviewSeer* has dramatically lower accuracy: only 38% (down from 88.4%). The accuracy was improved to 68% after removing difficult cases (called *I class*) and using only clearly positive or negative sentences about the given subject. *I class* includes sentences that were ambiguous when taken out of context (*case i*), were not describing

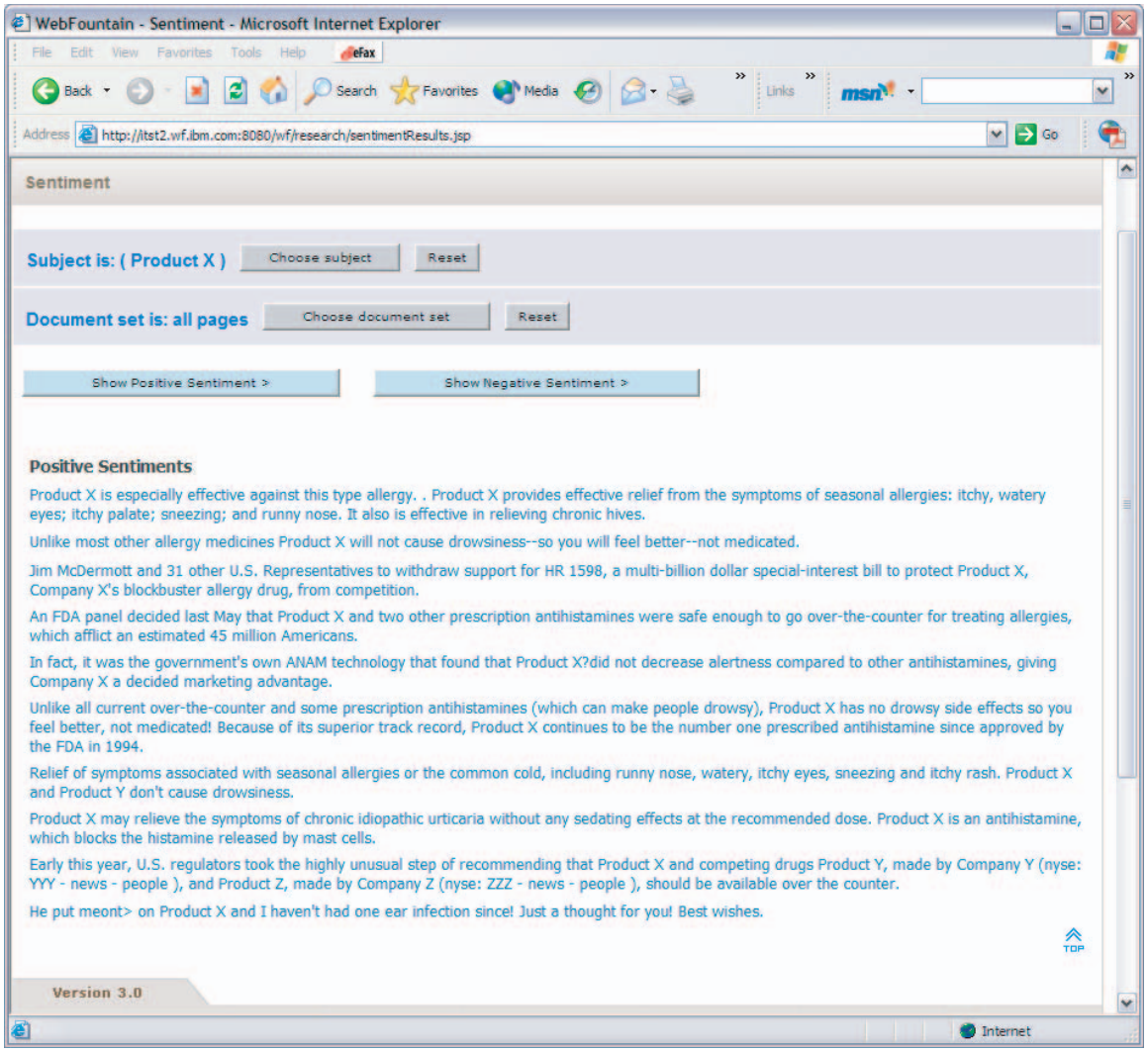

Figure 5. WebFountain Web interface listing sentiment bearing sentences for a given product.

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

Table 5. The performance of the sentiment miner and *ReviewSeer* on general web documents and news articles.

the product (*case ii*), or did not express any sentiment at all (*case iii*).

Note that these difficult cases are the majority of the sentiment bearing sentences: 60% $\sim$ 90% depending on the domain [6, 32] *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. The sentiment miner improved the (subject, sentiment) association accuracy by the semantic relationship analysis. The sentiment miner handles the neutral cases (case *iii*) already very well as discussed earlier.

Figure 4 and 5 are the GUI visualizations of the sentiment mining results on general web pages of pharmaceutical domain. The page contents are post-processed to mask out actual product names.

5 Results and Discussion

In this paper we described the sentiment mining system and algorithms and the WebFountain text analytics platform that the sentiment miner is running on. The sentiment mining prototype system currently runs and supports both re-

search and a set of customers who are involved in live use of applications hosted in the production environment.

Our NLP based sentiment mining system consistently achieved high quality results (∼90% of accuracy) on various datasets including online review articles and the general web pages and news articles. The results on general web documents are significantly better than those of the state of the art algorithms by a wide margin (38% vs. 91 ∼ 93%). The feature extraction algorithm successfully identified topic related feature terms from online review articles, enabling sentiment analysis at finer granularity.

References

[1] R. Agrawal, R. Bayardo, D. Gruhl, and S. Papadimitriou. Vinci: A service-oriented architecture for rapid development of web applications. In *Proceedings of the WWW Conference*, pages 355–365, 2002.

[2] E. Amitay, R. Nelken, W. Niblack, R. Sivan, and A. Soffer. Multi-resolution disambiguation of term occurrences. In *Proceedings of the ACM CIKM Conference*, 2003.

[3] Z. Bar-Yossef and S. Rajagopalan. Template detection via data mining and its applications. In *Proceedings of the International WWW Conference*, 2002.

[4] P. Beineke, T. Hastie, and S. Vaithyanathan. The sentimental factor: Improving review classification via human-provided information. In *Proceedings of the ACL Conference*, pages 263–270, Barcelona, Spain, 2004.

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

[6] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the Int. WWW Conference*, 2003.

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

[8] D. Gruhl, L. Chavet, D. Gibson, J. Meyer, P. Pattanayak, A. Tomkins, and J. Zien. How to build a webfountain: an architecture for very large-scale text analytics. *IBM Systems Journal*, 43(1):64–77, 2004.

[9] V. Hatzivassiloglou and K. R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the ACL Conference*, pages 174–181, 1997.

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

[11] A. Huettner and P. Subasic. Fuzzy typing for document management. In *3Proceedings of the ACL Conference (Software Demonstration)*, 2000.

[12] H. Li and K. Yamanishi. Mining from open answers in questionnaire data. In *Proceedings of the ACM SIGKDD Conference*, 2001.

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

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

[15] K. S. McCurley. Geospatial mapping and navigation of the web. In *Proceedings of the WWW Conference*, 2002.

[16] 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.

[17] S. Morinaga, K. Yamanishi, K. Teteishi, and T. Fukushima. Mining product reputations on the web. In *Proceedings of the ACM SIGKDD Conference*, 2002.

[18] T. Mullen and N. Collier. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the ACL Conference*, 2004.

[19] K. Nigam and M. Hurst. Towards a robust metric of opinion. In *Proceedings of the AAAI Spring Symposium Series on Exploring Attitude and Affect in Text: Theories and Applications*, 2004.

[20] B. Pang and L. Lee. A sentiment education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL Conference*, 2004.

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

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

[23] 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.

[24] W. Sack. On the computation of point of view. In *Proceedings of the AAAI Conference*, 1994.

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

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

[27] J. A. Tomlin. A new paradigm for ranking pages on the world wide web. In *Proceedings of the WWW Conference*, 2003.

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

[29] P. D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the ACL Conference*, pages 417–424, 2002.

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

[31] J. M. Wiebe. Learning subjective adjectives from corpora. In *Proceedings of the AAAI Conference*, 2000.

[32] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of the IEEE International Conference on Data Mining*, 2003.