

---

# A Novel Lexicalized HMM-based Learning Framework for Web Opinion Mining

---

**Wei Jin**

WEI.JIN@NDSU.EDU

Department of Computer Science, North Dakota State University, USA

**Hung Hay Ho**

HUNGHO@BUFFALO.EDU

Department of Computer Science & Engineering, State University of New York at Buffalo, USA

## Abstract

Merchants selling products on the Web often ask their customers to share their opinions and hands-on experiences on products they have purchased. As e-commerce is becoming more and more popular, the number of customer reviews a product receives grows rapidly. This makes it difficult for a potential customer to read them to make an informed decision on whether to purchase the product. In this research, we aim to mine customer reviews of a product and extract highly specific product related entities on which reviewers express their opinions. Opinion expressions and sentences are also identified and opinion orientations for each recognized product entity are classified as positive or negative. Different from previous approaches that have mostly relied on natural language processing techniques or statistic information, we propose a novel machine learning framework using lexicalized HMMs. The approach naturally integrates linguistic features, such as part-of-speech and surrounding contextual clues of words into automatic learning. The experimental results demonstrate the effectiveness of the proposed approach in web opinion mining and extraction from product reviews.

hundreds or even thousands. Our goal in this research is to design a framework that is capable of extracting, learning and classifying product related entities from product reviews.

Specifically, given a particular product, the system first identifies potential product related entities and opinion related entities from the reviews, and then extracts opinion sentences which describe each identified product entity, and finally determines opinion orientations (positive or negative) for each recognized product entity. Different from previous approaches that have mostly relied on natural language processing techniques (Turney, 2002) or statistic information (Hu and Liu, 2004), we propose a novel framework naturally integrates linguistic features (e.g., part-of-speech, phrases' internal formation patterns, and surrounding contextual clues of words/phrases) into automatic learning supported by lexicalized HMMs. The experimental results demonstrate the effectiveness of the proposed approach in web opinion mining and extraction from product reviews.

The rest of this paper is organized as follows: section 2 discusses background knowledge and related work. Section 3 describes each component of the proposed learning framework. We report in section 4 our experimental results and give our conclusions on this work in section 5.

## 1. Introduction

With the rapid expansion of e-commerce, more and more products are sold on the web, and more and more people are buying products online. It has become a common practice for online merchants to ask their customers to share their opinions and hands-on experiences on products they have purchased. Unfortunately, reading through all customer reviews is difficult, especially for popular items, the number of reviews can be up to

## 2. Related Work

Opinion mining can be divided into two categories, document level and feature level. Document level aims to classify the overall sentiment orientation of a document (Turney and Littman, 2003); feature level is interested in finding product features being commented on and the opinion polarity for each feature. In this paper, we are focused on feature level opinion mining and propose it involves two major tasks, recognition and classification. Recognition is the task of recognizing sentences expressing opinions; classification is the task of classifying elements in an opinion sentence into different categories such as opinion words/phrases and product features. Determining the polarity of opinion words (positive or negative) is also a classification task.

There has been work on feature level opinion mining. (Zhuang et al., 2006) classified and summarized movie reviews by extracting high frequency feature keywords and high frequency opinion keywords. Feature-opinion pairs were identified by using a dependency grammar graph. However, it used a fixed list of keywords to recognize high frequency feature words, and thus the system capability is limited. (Popescu and Etzioni, 2005) proposed a relaxation labeling approach to find the semantic orientation of words. However, their approach only extracted feature words with frequency greater than an experimentally set threshold value and ignored low frequency feature words. (Hu and Liu, 2004) proposed a statistical approach capturing high frequency feature words by using association rules. Infrequent feature words are captured by extracting known opinion words' adjacent noun phrases. A summary is generated by using high frequency feature words and ignoring infrequent features. (Ding et al., 2008) further improved Hu's system by manually adding some rules to handle different kinds of sentence structures. However, the capability of recognizing phrase features is limited by the accuracy of recognizing noun-group boundaries. Their approach also lacks an effective way to address infrequent features. In this work, we propose a novel framework naturally integrates linguistic features into automatic learning. The system can identify complex product-specific features (which are possible low frequency phrases in the reviews). The system can also self-learn new vocabularies based on the patterns it has seen from the training data. Therefore, the system is able to predict potential features in the test dataset even without seeing them in the training set. These capabilities were not supported by previous approaches.

### 3. The Proposed Techniques

Lexicalized HMMs was previously used in Part-of-Speech (POS) Tagging and Named Entity Recognition (NER) problem. The task of POS tagging is the process of marking up the words in a text (corpus) as corresponding to a particular part-of-speech, such as *noun* and *verb*. The task of NER is identifying and classifying person names, location names, organization names, and etc. To correlate the web opinion mining task with POS Tagging and NER may well be a significant contribution in itself in this work. We have adapted the techniques proposed in (Lee et al., 2000; Fu and Luke, 2005) for Korean part-of-speech tagging and Chinese named entity tagging respectively to better suit our task. Figure 1 gives the architectural overview of our opinion mining system. Below, we discuss each of the sub-steps in turn.

#### 3.1 Entity Categories and Tag Sets

In our work, we have defined four entity categories as shown in table 1 (a digital camera is used as an example).

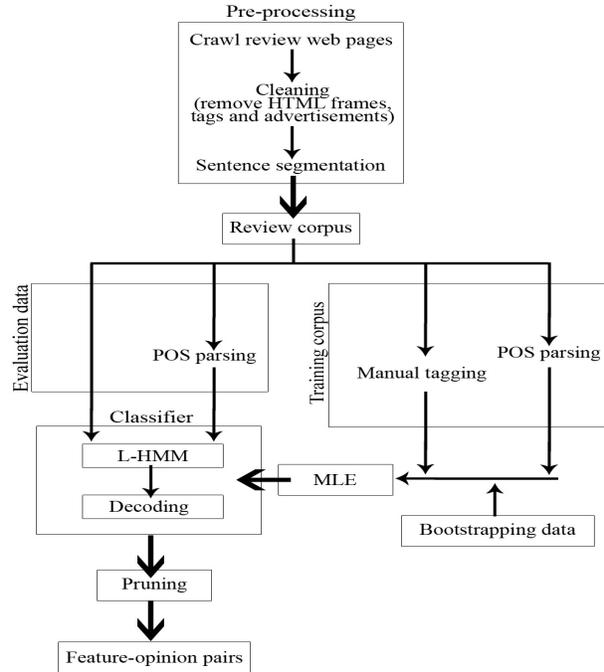


Figure 1. The system framework

Table 1. Definitions of entity categories and examples

COMPONENTS	Physical objects of a camera including the camera itself, e.g., LCD, viewfinder, battery
FUNCTIONS	Capabilities provided by a camera, e.g., movie playback, zoom, automatic fill-flash, auto focus
FEATURES	Properties of components or functions, e.g., color, speed, size, weight, clarity
OPINIONS	Ideas and thoughts expressed by reviewers on product features / components / functions.

Correspondingly, we have further defined the basic tag set to identify each above entity category, which is given in table 2.

In general, an entity can be a single word or a phrase. In other words, a word may present itself as an independent entity or a component of an entity. Therefore, a word  $w$  in an entity may take one of the following four patterns to present itself:

1.  $w$  is an independent entity;
2.  $w$  is the beginning component of an entity;
3.  $w$  is at the middle of an entity;
4.  $w$  is at the end of an entity.

We use a pattern tag set to denote the above four patterns, which is given in table 3.

Table 2. Basic tag set and its corresponding entities

TAGS	CORRESPONDING ENTITIES
<PROD_FEAT>	Feature Entities
<PROD_COMP>	Component Entities
<PROD_FUNCTION>	Function Entities
<OPINION_POS_EXP>	Explicit Positive Opinion Entities
<OPINION_NEG_EXP>	Explicit Negative Opinion Entities
<OPINION_POS_IMP>	Implicit Positive Opinion Entities
<OPINION_NEG_IMP>	Implicit Negative Opinion Entities
<BG>	Background Words

Table 3. Pattern tag set and its corresponding patterns

PATTERN TAGS	CORRESPONDING PATTERNS
<>	Independent Entities (Single Words)
<BOE>	The Beginning Component of an Entity
<MOE>	The Middle Component of an Entity
<EOE>	The End of an Entity

Both the basic tag set and pattern tag set are used to represent each word's entity category and pattern (referred to as a hybrid tag representation as proposed in (Fu and Luke, 2005)). Patterns of background words are considered as *independent entities*. This hybrid-tag labeling method is applied to all the training data and system outputs.

The following example illustrates the hybrid tag and basic tag representations of an opinion sentence “*I love the ease of transferring the pictures to my computer.*”

Hybrid tags:

<BG>I</BG><OPINION\_POS\_EXP>love</OPINION\_POS\_EXP><BG>the</BG><PROD\_FEAT-BOE>ease</PROD\_FEAT-BOE> <PROD\_FEAT-MOE>of</PROD\_FEAT-MOE><PROD\_FEAT-MOE>transferring</PROD\_FEAT-MOE> <PROD\_FEAT-MOE>the</PROD\_FEAT-MOE> <PROD\_FEAT-EOE>pictures</PROD\_FEAT-EOE> <BG>to</BG><BG>my</BG><BG>computer</BG>

Basic tags:

<BG>I</BG><OPINION\_POS\_EXP>love</OPINION\_POS\_EXP> <BG> the </BG> <PROD\_FEAT>ease of transferring the pictures </PROD\_FEAT> <BG> to </BG><BG>my</BG><BG>computer</BG>

### 3.2 Lexicalized HMMs

Different from traditional Hidden Markov Models, in our work, we integrate linguistic features such as part-of-speech and lexical patterns into HMMs. An observable state is represented by a pair  $(word_i, POS(word_i))$  where  $POS(word_i)$  represents the part-of-speech of  $word_i$ .

The task is then described as follows: Given a sequence of words  $W = w_1 w_2 w_3 \dots w_n$  and corresponding parts-of-speech  $S = s_1 s_2 s_3 \dots s_n$ , the task is to find an appropriate sequence of hybrid tags  $\hat{T} = t_1 t_2 t_3 \dots t_n$  that maximize the conditional probability  $P(T|W, S)$ , namely

$$\hat{T} = \arg \max_T P(T|W, S) = \arg \max_T \frac{P(W, S|T)P(T)}{P(W, S)} \quad (1)$$

Since the probability  $P(W, S)$  remains unchanged for all candidate tag sequences, we can disregard it. Thus, we have a general statistical model as follows:

$$\begin{aligned} \hat{T} &= \arg \max_T P(W, S|T)P(T) = \arg \max_T P(S|T)P(W|T, S)p(T) \\ &= \arg \max_T \prod_{i=1}^n \left( \begin{array}{l} P(s_i | w_1 \dots w_{i-1}, s_1 \dots s_{i-1}, t_1 \dots t_{i-1}) \times \\ P(w_i | w_1 \dots w_{i-1}, s_1 \dots s_{i-1}, s_i, t_1 \dots t_{i-1}) \times \\ P(t_i | w_1 \dots w_{i-1}, s_1 \dots s_{i-1}, t_1 \dots t_{i-1}) \end{array} \right) \end{aligned} \quad (2)$$

Theoretically the general model can provide the system with a powerful capacity of disambiguation. However, in practice this general model is not computable for it involves too many parameters. Two types of approximations are made to simplify the general model.

The first approximation is based on the independent hypothesis used in standard HMMs. First-order HMMs is used in view of data sparseness, i.e.,

$$P(t_i | t_{i-K} \dots t_{i-1}) \approx P(t_i | t_{i-1}).$$

The second approximation combines the POS information with the lexicalization technique where three main hypotheses are made:

1. The assignment of current tag  $t_i$  is supposed to depend not only on its previous tag  $t_{i-1}$  but also previous  $J$  ( $1 \leq J \leq i-1$ ) words  $w_{i-J} \dots w_{i-1}$ .
2. The appearance of current word  $w_i$  is assumed to depend not only on the current tag  $t_i$ , current POS  $s_i$ , but also the previous  $K$  ( $1 \leq K \leq i-1$ ) words  $w_{i-K} \dots w_{i-1}$ .
3. The appearance of current POS  $s_i$  is supposed to depend both on the current tag  $t_i$  and previous  $L$  ( $1 \leq L \leq i-1$ ) words  $w_{i-L} \dots w_{i-1}$ .

With a view to the issue of data sparseness, we set  $J=K=L=1$ . Based on these assumptions, the general model in equation (2) can be rewritten as:

$$\hat{T} = \arg \max_T \prod_{i=1}^n \left( \begin{array}{l} P(s_i | w_{i-1}, t_i) \times \\ P(w_i | w_{i-1}, s_i, t_i) \times \\ P(t_i | w_{i-1}, t_{i-1}) \end{array} \right) \quad (3)$$

Maximum Likelihood Estimation (MLE) is used to estimate the parameters in equation (3). For instance,  $P(s_i | w_{i-1}, t_i)$  can be estimated as:

$$P(s_i | w_{i-1}, t_i) = \frac{C(w_{i-1}, t_i, s_i)}{\sum_s C(w_{i-1}, t_i, s)} = \frac{C(w_{i-1}, t_i, s_i)}{C(w_{i-1}, t_i)} \quad (4)$$

Note that the sum of counts of  $C(w_{i-1}, t_i, s)$  for all  $s$  is equivalent to the count of  $C(w_{i-1}, t_i)$ . MLE values for other estimations in equation (3) can be computed similarly.

Combing the lexicalization technique with POS information is able to handle richer contextual information for the assignment of tags to known words, including both contextual words and contextual tags under the framework of HMMs. Consequently, the accuracy of the recognizer can be improved without losing its efficiency in training and tagging. If a large training corpus is available, the parameters in equation (3) can be easily estimated. However, MLE will yield zero probabilities for any cases that are not observed in the training data. To solve this problem, we employ the linear interpolation smoothing technique to smooth higher-order models with their relevant lower-order models, or to smooth the lexicalized parameters using the related non-lexicalized probabilities, namely

$$\begin{aligned} P'(s_i | w_{i-1}, t_i) &= \lambda P(s_i | w_{i-1}, t_i) + (1 - \lambda) P(s_i | t_i) \\ P'(w_i | w_{i-1}, s_i, t_i) &= \left( \frac{\beta P(w_i | w_{i-1}, s_i, t_i) +}{(1 - \beta) P(w_i | s_i, t_i)} \right) \quad (5) \\ P'(t_i | w_{i-1}, t_{i-1}) &= \alpha P(t_i | w_{i-1}, t_{i-1}) + (1 - \alpha) P(t_i | t_{i-1}) \end{aligned}$$

Where  $\lambda$ ,  $\beta$  and  $\alpha$  denote the interpolation coefficients (We observed that large values of  $\lambda$ ,  $\beta$  and  $\alpha$  achieved high precision but low recall and low values of  $\lambda$ ,  $\beta$  and  $\alpha$  achieved low precision but high recall. In terms of F-score, the settings 0.7 for  $\lambda$ ,  $\beta$  and  $\alpha$  achieved the best performance in our experiments).

### 3.3 Tagging

Based on the above models, the tagging algorithm aims at finding the most probable sequence of hybrid tags for a given sequence of known words and corresponding parts-of-speech. The algorithm is implemented in three major steps as follows:

1. The generation of candidate tags: This step aims to generate candidate hybrid tags for a sequence of known words and corresponding parts-of-speech. As discussed above, the candidate hybrid tags of a known word are a combination of its candidate category tags and its candidate pattern tags.
2. The decoding of the best tag sequence: In this step, the Viterbi algorithm is employed to score all candidate hybrid tags with the proposed model, and then search the best path that has the maximal score.

3. The conversion of the results: For evaluation purposes, the hybrid-tag output produced by the tagger is further converted to the basic-tag format (the example is shown in section 3.1) by merging the consecutive known words into entities in terms of their patterns.

### 3.4 Opinion Sentence Extraction

This step identifies opinion sentences in the reviews. Opinion sentences in our work are defined as sentences that express an opinion on product related entities. In the pruning step, the following two types of sentences are not considered as effective opinion sentences:

1. Sentences that describe product related entities without expressing reviewers' opinions.
2. Sentences that express opinions on another product model's entities (model numbers, such as DMC-LS70S, P5100, A570IS, can be easily identified using the regular expression "[A-Z-]+\d+[A-Za-z+]?").

### 3.5 Determining Opinion Orientation

This step further classifies opinion orientation given each identified product entity. Due to the complexity of natural language, opinion orientation is not simply equal to opinion entity (word/phrase)'s orientation. For example,

*"I can tell you right now that the auto mode and the program modes are not that good."*

The reviewer expressed his negative comment on both "auto mode" and "program modes" even in the presence of the opinion entity (word "good") in the sentence.

To determine opinion orientation, we first convert the hybrid tagged sentences to basic tagged sentences, and then for each recognized product entity, we search its matching opinion entity, which is defined as the nearest opinion word/phrase identified by the tagger. The orientation of this matching opinion entity becomes the initial opinion orientation for the corresponding product entity. Next, natural language rules reflecting the sentence context are employed to deal with specific language constructs which may change the opinion orientation, such as the presence of negation words (e.g., not). The details of implementation are shown in Algorithm 1.

Line 8 to line 23 checks the presence of any negation words (e.g., not, didn't, don't) within five-word distance in front of an opinion word/phrase and changes opinion orientation accordingly, except

1. A negation word appears in front of a coordinating conjunction (e.g., and, or, but). (line 10 – 13)
2. A negation word appears after the appearance of a product entity during the backward search within the five-word window. (line 14 -17)

Line 27 to 32 handles the coordinating conjunction “but” and prepositions such as “except” and “apart from”. The purpose of this procedure is to resolve the true opinion polarity for product entities when:

1. The opinion expression in the “but” clause is unsolvable, e.g., “It takes great movies with sound, but the sound can not be played back on the camera, only when it is connected to a TV or after you upload them.” Reviewers expressed a negative comment on the camera’s soundless movie playback.
2. The opinion is expressed on a bunch of product entities except some. For instance, “Everything about the camera is great, except they only went with a 2 battery system.”

---

**Algorithm 1** Determining Opinion Orientation

---

```

RESOLVE_OPINION_ORI(tagged OpinionSentence)
1. FOR each product related entity  $f_i$  in OpinionSentence
2.   corresponding opinion entity  $o_i = f_i$ 's matching
3.   opinion word/phrase
4.    $f_i$ 's initial opinion orientation =  $o_i$ 's orientation
5.
6.   // look backwards and search for negation words
7.   done = FALSE
8.   FOR (distance = 1; distance <= 5 && !done;
9.   distance++)
10.    IF (( $o_i$ 's position - distance) is a coordinating
11.    conjunction)
12.      done = TRUE
13.    END IF
14.    IF (( $o_i$ 's position - distance) is in front of  $f_i$ 's
15.    position)
16.      done = TRUE
17.    END IF
18.    IF (( $o_i$ 's position - distance) is a negation word)
19.      done = TRUE
20.       $f_i$ 's opinion orientation = opposite( $f_i$ 's initial
21.      opinion orientation)
22.    END IF
23.  END FOR
24.
25. // handling the conjunctions such as “but” and
26. prepositions such as “except”
27. IF  $o_i$  is in front of  $f_i$ 
28.   IF “but/except” appears between  $o_i$  and  $f_i$ 
29.     $f_i$ 's opinion orientation = opposite( $f_i$ 's initial
30.    opinion orientation)
31.   END IF
32. END IF
33. END FOR

```

---

## 4. Experiments

We used Amazon’s digital camera reviews as the evaluation dataset. The reviews for the first 16 unique cameras listed on Amazon.com during November 2007 were crawled. For each review page, each individual review content, model number as well as manufacturer name were extracted from the HTML documents. Sentence segmentation was applied to the data and the information was stored as plain text documents, which we call *review documents*. POS parsing was applied to each review document. We used the Part-of-Speech tagger designed by the Stanford NLP Group<sup>1</sup> and default settings of the tagger were used.

### 4.1 Training Design

After downloading and pre-processing, there were 1728 review documents obtained. We separated the documents into 2 sets. One set (293 documents for 6 cameras) were manually tagged by experts. For each opinion sentence, product entities, opining entities and opinion orientations were manually labeled using the tag sets described in section 3.1. The remaining documents (1435 documents for 10 cameras) were used by the bootstrapping process to self-learn new vocabularies (described next). Additionally, we observed two challenges in this task. One was people use inconsistent terminologies to describe product entities, and the other arose from recognizing rarely mentioned entities (i.e. infrequent entities). For example, only two reviewers mentioned “poor design on battery door” in the training data. Although this is not a frequent entity, it provides valuable information to potential customers. Again, different terms were used to describe it, such as battery cover, battery/SD door, battery/SD cover, battery/SD card cover. To reduce the effort of manual labeling of a large set of training documents and solve the above problems, we propose a novel bootstrapping approach to enable self-directed learning, which can be employed in situations where collecting a large training set could be expensive and difficult to accomplish.

### 4.2 Bootstrapping

Labeling training documents manually is a labor intensive task. Thus, it would be nice if the system can discover new vocabularies automatically by using what it has learned. To achieve this, we have designed a bootstrapping approach which can extract high confidence data through self-learning. The process is shown in Fig. 2 and composed of the following steps:

1. First, the bootstrapping program creates two child processes. The parent process acts as master and the rest acts as workers. Master is responsible for coordinating the bootstrapping process, extracting and distributing high confidence data to each worker.

---

<sup>1</sup> <http://nlp.stanford.edu/software/lex-parser.shtml>

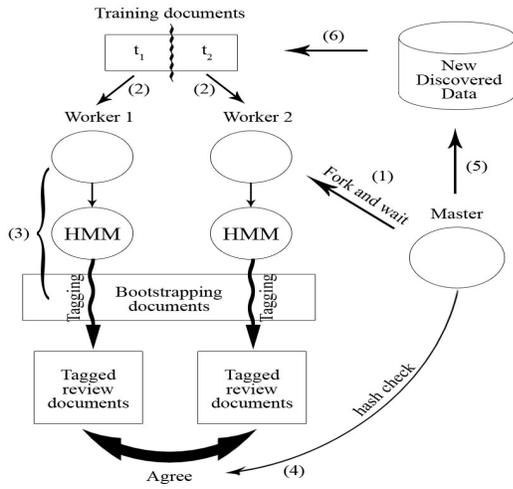


Figure 2. The bootstrapping process

2. We split the training documents into two halves,  $t_1$  and  $t_2$  by random selection. Each half is used as seeds for each worker's HMM.
3. At the initial stage (0<sup>th</sup> iteration), each worker trains its own HMM classifier based on its training set, and then each worker's trained HMM is used to tag the documents in the bootstrap document set and produces a new set of tagged review documents.
4. As two workers' training documents are different from each other, the tagging results from step 3 may be inconsistent. Therefore after the tagging step, master inspects each sentence tagged by each HMM classifier and only extracts opinion sentences that are agreed upon by both classifiers. In the experiments, only the identical sentences with identical tags were considered to agree with one another.
5. A hash value is then calculated for each extracted opinion sentence from step 4 and compared with those of the sentences already stored in the database (The database contains newly discovered data from the bootstrap process and is initialized to empty in the first bootstrap cycle). If it is a newly discovered sentence, master stores it into the database.
6. Master then randomly splits the newly discovered data from the database into two halves  $t_1$  and  $t_2$ , and adds  $t_1$  and  $t_2$  to the training set of two workers respectively. This bootstrap process is repeated until no more new data being discovered. Figure 3 and 4 demonstrate the experimental results obtained from each bootstrap cycle regarding one of the products used in our experiments.

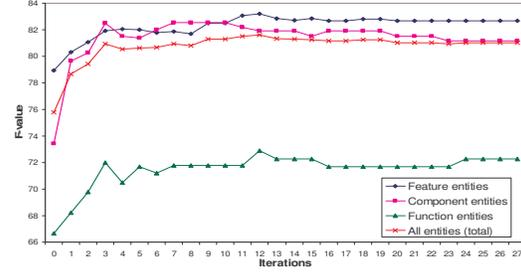


Figure 3. Bootstrapping results for entity extraction

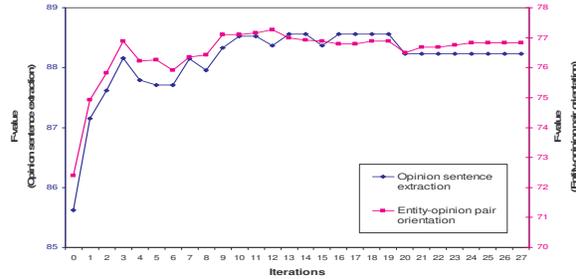


Figure 4. Bootstrapping results for opinion extraction and entity-opinion pair orientation

### 4.3 Evaluation

As mentioned above, the review documents for 6 cameras were manually labeled by experts. We chose the largest four data sets (containing 270 documents) and performed a 4-fold cross-validation. The remaining review documents for 2 cameras (containing 23 documents) were used for training only. The bootstrap document set (containing 1435 documents for 10 cameras) was used by the bootstrapping process to extract high confidence data through self-learning (newly discovered high confidence data were then added into the original training set in each iteration). Finally, our best classifier was trained based on the accumulated truth data collected from the original training set and bootstrap data set, and was then applied to our test data and evaluated against the baseline.

To evaluate the effectiveness of the proposed framework, we have measured the recall, precision and F-score of extracted entities, opinion sentences and opinion orientations, respectively. The system performance is evaluated by comparing the results tagged by the system with the manually tagged truth data. Only an exact match is considered as a correct recognition in our evaluation. For entity recognition, this means the exact same word/phrase is identified and classified correctly from one of four pre-defined entity categories. Furthermore, each identified entity should occur in the same sentence, same position and same document as compared with the truth data. For opinion sentence extraction, exact match means

the exact same sentence from the same document is identified compared with the truth data. For opinion orientation classification, exact match means the exact same entity and entity category are identified with correct orientation (positive or negative).

We have designed and implemented a rule-based baseline system motivated by (Turney 2002) and (Hu and Liu 2004)'s approaches. (Turney 2002) described a document level opinion mining system. It uses a number of rules to identify opinion-bearing words. In our baseline system, the rules shown in Table 4 were used to extract product entities and opinion-bearing words. This was accomplished by searching for any nouns and adjectives matching the rules. Matching nouns (considered as product entities) and matching adjectives (considered as opinion words) were extracted. The corresponding sentences were identified as opinion sentences. In the next step, identified adjectives' semantic orientations were determined. We used twenty five commonly used positive adjectives and twenty five commonly used negative adjectives as seeds. By using the bootstrapping technique proposed in (Hu and Liu 2004), we expanded these two seeds lists by searching synonyms and antonyms for each seed word. Newly discovered words were added into their corresponding seeds lists. This process was repeated until no new word discovered. As semantic orientation of each list of adjective words is known, the orientations of extracted adjectives by the system can be determined by checking the existence of these words in the lists.

The detailed evaluation results are presented in Table 5, 6 and 7. As a post analysis, the proposed machine learning approach performs significantly better than the rule-based baseline system in terms of entity extraction, opinion sentence recognition and opinion polarity classification. Through manual inspection, we observe the approach effectively identifies highly specific product related entities and opinion expressions (usually complex phrases, e.g., auto red eye correction) and discovers new vocabularies (e.g., automatic point-and-shoot mode) in the test dataset based on the patterns it has learned.

Table 4. Baseline rules for extracting product entities and opinion-bearing words

	FIRST WORD	SECOND WORD	THIRD WORD
1	JJ	NN or NNS	Anything
2	RB, RBR or RBS	JJ	NN or NNS
3	JJ	JJ	NN or NNS
4	NN or NNS	JJ	Not NN nor NNS (not extracted)

## 5. Conclusions

This paper proposes a novel and robust machine learning approach for web opinion mining and extraction. The

model provides solutions for several problems that have not been addressed by previous approaches. Specifically, the system can self-learn new vocabularies based on the patterns it has learned, which is extremely useful in text and web mining due to the complexity and flexibility of natural language and was not supported by previous rule-based or statistical approaches. Complex product entities and opinion expressions as well as infrequently mentioned entities can be effectively and efficiently identified, which was under-analyzed or ignored by previously proposed methods. A novel bootstrapping approach is employed to handle situations in which collecting a large training set could be expensive and difficult to accomplish.

Future directions include the expansion of the datasets from digital product reviews to other product reviews. We are also researching the role of pronoun resolution in improving the mining results.

## References

- Ding, X, Liu, B and Yu, P. S. (2008). A Holistic Lexicon-based Approach to Opinion Mining. *Proceedings of the International Conference on Web Search and Web Data Mining (WSDM'08)*, 231-239.
- Fu, G. and Luke, K. K. (2005). Chinese Named Entity Recognition using Lexicalized HMMs. *ACM SIGKDD Explorations Newsletter*, 7(1), 19-25.
- Hu, M. and Liu, B. (2004). Mining and Summarizing Customer Reviews. *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, 168-177.
- Lee, S. Z., Tsujii, J. and Rim, H. C. (2000). Lexicalized Hidden Markov Models for Part-of-Speech Tagging. *Proceedings of the 18th International Conference on Computational Linguistics (COLING'00)*, 481-487.
- Popescu, A. and Etzioni, O. (2005). Extracting Product Features and Opinions from Reviews. *Proceedings of 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP'05)*, 339-346.
- Turney, P. D. (2002). Thumbs up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, 417-424.
- Turney, P. D. and Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association." *ACM Transactions on Information Systems*, 21(4), 315-346.
- Zhuang, L., Jing, F. and Zhu, X. (2006). Movie Review Mining and Summarization. *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM'06)*, 43-50.

Table 5. Experimental results on entity extraction for each category

PRODUCTS	METHODS	FEATURE ENTITY			COMPONENT ENTITY			FUNCTION ENTITY		
		R(%)	P(%)	F(%)	R(%)	P(%)	F(%)	R(%)	P(%)	F(%)
CAMERA A	L-HMM+POS+Bootstrapping	85.81	80.78	83.22	83.33	83.33	83.33	70.31	75.00	72.58
	L-HMM+POS	82.01	77.70	79.80	73.08	73.08	73.08	65.63	70.00	67.74
	L-HMM	80.74	75.86	78.22	70.36	70.30	70.33	60.19	67.19	63.50
CAMERA B	L-HMM+POS+Bootstrapping	89.12	75.72	81.88	77.66	79.35	78.49	60.87	82.35	70.00
	L-HMM+POS	84.35	72.09	77.74	74.47	70.92	72.65	52.17	75.00	61.54
	L-HMM	80.67	71.51	75.81	71.66	70.46	71.05	47.83	64.71	55.00
CAMERA C	L-HMM+POS+Bootstrapping	76.62	81.94	79.19	100.0	83.67	91.11	63.64	87.50	73.69
	L-HMM+POS	74.03	80.28	77.03	97.56	78.43	86.96	63.64	80.50	71.08
	L-HMM	70.32	77.33	73.66	97.56	74.07	84.21	63.64	70.00	66.67

Table 6. Experimental results on entity extraction for all categories

PRODUCTS	METHODS	ALL ENTITIES (TOTAL)		
		R(%)	P(%)	F(%)
CAMERA A	L-HMM+POS+Bootstrapping	83.10	80.88	81.98
	L-HMM+POS	77.21	75.43	76.31
	L-HMM	75.78	73.18	74.46
	Baseline	20.43	29.97	24.30
CAMERA B	L-HMM+POS+Bootstrapping	82.58	77.30	79.85
	L-HMM+POS	78.03	71.84	74.81
	L-HMM	74.81	70.87	72.78
	Baseline	15.53	24.26	18.94
CAMERA C	L-HMM+POS+Bootstrapping	82.95	82.95	82.95
	L-HMM+POS	80.62	79.60	80.11
	L-HMM	78.23	75.54	76.86
	Baseline	17.05	23.66	19.82

Table 7. Experimental results on opinion sentence identification and opinion orientation classification

PRODUCTS	METHODS	OPINION SENTENCE EXTRACTION (SENTENCE LEVEL)			ENTITY-OPINION PAIRS ORIENTATION (FEATURE LEVEL)		
		R(%)	P(%)	F(%)	R(%)	P(%)	F(%)
CAMERA A	L-HMM+POS+Bootstrapping	90.72	85.71	88.15	78.98	76.86	77.91
	L-HMM+POS	87.63	83.88	85.71	74.26	72.55	73.40
	L-HMM	86.32	82.11	84.16	73.25	69.89	71.53
	Baseline	51.89	60.64	55.93	19.65	28.82	23.36
CAMERA B	L-HMM+POS+Bootstrapping	87.95	82.95	85.38	75.00	70.21	72.53
	L-HMM+POS	86.75	81.82	84.21	69.70	65.95	67.77
	L-HMM	85.14	80.29	82.64	68.45	65.02	66.69
	Baseline	46.39	57.04	51.56	13.26	20.71	16.17
CAMERA C	L-HMM+POS+Bootstrapping	83.91	82.95	83.43	77.52	77.52	77.52
	L-HMM+POS	80.46	80.34	80.40	72.87	72.31	72.59
	L-HMM	79.76	78.82	79.29	72.09	66.91	69.40
	Baseline	43.68	54.29	48.41	17.05	23.66	19.82