OpinionMiner: A Novel Machine Learning System for Web Opinion Mining and Extraction

Wei Jin
Department of Computer Science
North Dakota State University
Fargo, ND 58108
wei.jin@ndsu.edu

Hung Hay Ho
Department of Computer Science & Engineering
State University of New York at Buffalo
Buffalo, NY 14260
hungho@buffalo.edu

Rohini K. Srihari
Department of Computer Science & Engineering
State University of New York at Buffalo
Buffalo, NY 14260
rohini@cedar.buffalo.edu

ABSTRACT
Merchants selling products on the Web often 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 hundreds or even thousands. This makes it difficult for a potential customer to read them to make an informed decision. The OpinionMiner system designed in this work aims to mine customer reviews of a product and extract high detailed product entities on which reviewers express their opinions. Opinion expressions are identified and opinion orientations for each recognized product entity are classified as positive or negative. Different from previous approaches that employed rule-based or statistical techniques, we propose a novel machine learning approach built under the framework of lexicalized HMMs. The approach naturally integrates multiple important linguistic features into automatic learning. In this paper, we describe the architecture and main components of the system. The evaluation of the proposed method is presented based on processing the online product reviews from Amazon and other publicly available datasets.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Information filtering*. I.2.7 [**Natural Language Processing**] – *Text analysis*

General Terms
Algorithms, Design, Experimentation, Human Factors

Keywords
Opinion Mining, Sentiment Analysis, Lexicalized HMMs

1. INTRODUCTION
As e-commerce is becoming more and more popular, 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. Such information is highly valuable to manufacturers, online advertisers and potential customers. Unfortunately, reading through all customer reviews is difficult, especially for popular items, the number of reviews can be up to hundreds or even thousands. This makes it difficult for a potential customer to read them to make an informed decision on whether to purchase the product. This paper aims to design a system that is capable of extracting, learning and classifying product entities and opinion expressions automatically from product reviews. A novel lexicalized HMM-based approach is proposed and an opinion mining and extraction system, *OpinionMiner*, has been developed. Our objective in this system is to answer the following questions: given a particular product, 1) how to automatically extract potential product entities and opinion entities from the reviews? 2) how to identify opinion sentences which describe each extracted product entity? and 3) how to determine opinion orientation (positive or negative) given each recognized product entity? Different from previous approaches that have mostly relied on rule-based techniques [3, 4] or statistic information [10, 13], we propose a new framework that naturally integrates multiple linguistic features (e.g., part-of-speech, phrases' internal formation patterns, and surrounding contextual clues of words/phrases) into automatic learning. The experimental results demonstrate the effectiveness of the proposed approach in web opinion mining and extraction from online product reviews.

Our contributions in this paper include: (1) a proposal of a new machine learning framework that naturally integrates multiple linguistic features into web opinion mining and extraction; (2) a proposal of a unified and self-adaptive tagging approach including use of dictionary, token transformations and bootstrapping; (3) a proposal of an effective approach of extracting complex product entities, opinion expressions, as well as infrequently mentioned entities from reviews; (4) a proposal of a practically effective system design.

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

KDD'09, June 28–July 1, 2009, Paris, France.

2. Related Work

Opinion analysis has been studied by many researchers in recent years. Two main research directions are explored, i.e., document level opinion mining and feature level opinion mining. In document level, Turney [3] presented an approach of determining document's polarity by calculating the average semantic orientation (SO) of extracted phrases. SO was computed by using pointwise mutual information (PMI) to measure the dependence between extracted phrases and the reference words "excellent" and "poor" by using web search hit counts. One year later Turney and Littman [4] further expanded their work by using cosine distance in latent semantic analysis (LSA) as the distance measure. Dave, Lawrence and Pennock [5] classified reviews on Amazon by calculating scores using normalized term frequency on uni-gram, bi-gram and tri-gram with different smoothing techniques. Das and Chen [8] studied document level sentiment polarity classification on financial documents. Pang, Lee and Vaithyanathan [6] used several machine learning approaches to classify movie reviews and in [7], they further studied another machine learning approach based on subjectivity detection and minimum cuts in graphs for sentiment classification of movie reviews. Our work is different from these as their goal is to determine the sentiment of documents while ours is to perform extraction and classification on entities. Another difference is they were not focused on features being commented on.

In feature level opinion mining, Zhuang, Jing and Zhu [10] 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 [11] 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 [9] 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 (the top ranked features) and ignoring infrequent features. Ding, Liu and Yu [12] further improved Hu's system by 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 new machine learning framework that naturally integrates linguistic features into automatic learning. Complex product-specific features (which are possible low frequency phrases in the reviews) are effectively identified, and new potential product and opinion entities are discovered based on the patterns the classifier has seen from the training data.

Another related research area is the Part-of-Speech (POS) Tagging and Named Entity Recognition (NER) problems. 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, organization names, and etc. In opinion mining, similar tasks need to be performed, such as identifying different entity names, classifying entities into appropriate categories, and further determining the opinion word/phrase's polarity. To correlate the web opinion mining task with POS Tagging and NER may well be a significant contribution in itself in this work.

3. THE PROPOSED FRAMEWORK

Motivated by [1, 2] which employed a lexicalized HMM approach in Korean part-of-speech (POS) tagging and Chinese named entity tagging respectively, we propose in this paper a hybrid approach integrating POS information with the lexicalization technique under the HMM framework. Figure 1 gives the architectural overview of our opinion mining system and each system component is detailed subsequently.

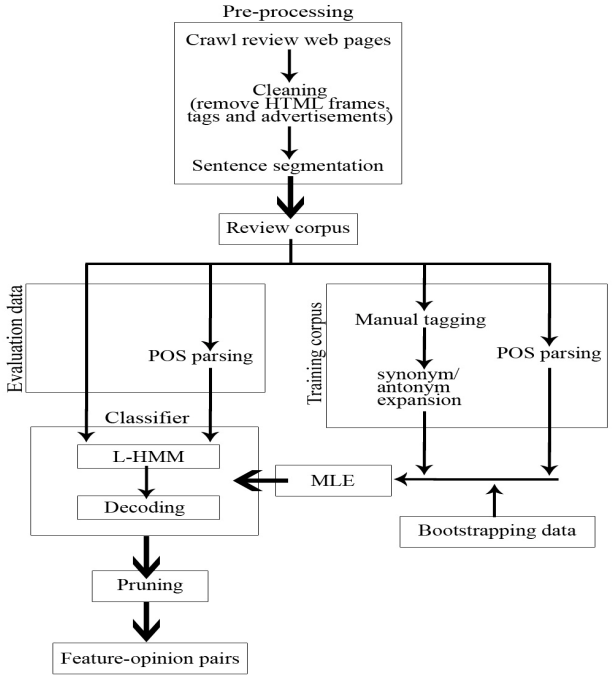

Figure 1. The System Framework

3.1 Entity Types and Tag Sets

In our work, we have defined four entity types as shown below (a digital camera is used as an example):

Table 1. Definitions of entity types 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 type, which is given below.

Table 2. Basic tag set and its corresponding entities

Tag set	Corresponding Entities
<PROD_FEAT>	Feature entity
<PROD_PARTS>	Component entity
<PROD_FUNCTION>	Function entity
<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

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 entity. Therefore, a word *w* in an entity may take one of the following four patterns to present itself: (i) *w* is an independent entity; (ii) *w* is the beginning component of an entity; (iii) *w* is at the middle of an entity; (iv) *w* is at the end of an entity. We adopt a pattern tag set proposed in [2] to denote the above four patterns, which is shown in table 3:

Table 3. Pattern tag set and its corresponding pattern

Pattern Tag	Corresponding Pattern
<>	Independent Entity
<-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 type and pattern (referred to as a hybrid tag representation [2]) in the following format: $t_b t_p$ where t_b represents a basic tag and t_p represents a pattern tag. Thus, an opinion sentence can be represented as

$<t_b t_p>w_1</t_b t_p>\ldots\ldots<t_b t_p>w_n</t_b t_p>$

where w_i stands for a single word.

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 Integrating POS

Different from traditional Hidden Markov Models (HMMs), 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 $W= w_1 w_2 w_3 \ldots w_n$ and corresponding parts of speech $S = s_1 s_2 s_3 \ldots s_n$, the task is to find an appropriate sequence of hybrid tags $\hat{T} = t_1 t_2 t_3 \ldots t_n$ that maximize the conditional probability $P(T|W,S)$ such that

$$\hat{T} = \arg\max_T P(T \mid W, S) \tag{1}$$

By taking Bayes law, we can rewrite equation (1) as

$$\hat{T} = \arg\max_T \frac{P(W, S \mid T) P(T)}{P(W, S)} \tag{2}$$

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:

$$\begin{aligned} \hat{T} &= \arg\max_T P(W, S \mid T) P(T) = \arg\max_T P(S \mid T) P(W \mid T, S) p(T) \\ &= \arg\max_T \prod_{i=1}^{n} \begin{pmatrix} P(s_i \mid w_1 \ldots w_{i-1}, s_1 \ldots s_{i-1}, t_1 \ldots t_{i-1} t_i) \times \\ P(w_i \mid w_1 \ldots w_{i-1}, s_1 \ldots s_{i-1} s_i, t_1 \ldots t_{i-1} t_i) \times \\ P(t_i \mid w_1 \ldots w_{i-1}, s_1 \ldots s_{i-1}, t_1 \ldots t_{i-1}) \end{pmatrix} \end{aligned} \tag{3}$$

Theoretically the above 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 employed to simplify this 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., $P(t_i \mid t_{i-K} \ldots t_{i-1}) \approx P(t_i \mid t_{i-1})$. The second approximation employs the lexicalization technique together with POS 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 $(1 \le J \le i-1)$ words $w_{i-J} \ldots w_{i-1}$.
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 $K(1 \le K \le i-1)$ words $w_{i-K} \ldots w_{i-1}$.
3. The appearance of current POS s_i is supposed to depend both on the current tag t_i and previous $L(1 \le L \le i-1)$ words $w_{i-L} \ldots w_{i-1}$.

With a view to the issue of data sparseness, we set $J=K=L=1$. Based on these assumptions, the general model in equation (3) can be rewritten as:

$$\hat{T} = \arg\max_{T} \prod_{i=1}^{n} \begin{pmatrix} P(s_i \mid w_{i-1}, t_i) \times \\ P(w_i \mid w_{i-1}, s_i, t_i) \times \\ P(t_i \mid w_{i-1}, t_{i-1}) \end{pmatrix} \quad (4)$$

Maximum Likelihood Estimation (MLE) is used to estimate the parameters in equation (4). For instance, $P(s_i|w_{i-1},t_i)$ can be estimated as:

$$P(s_i \mid w_{i-1}, t_i) = \frac{C(w_{i-1}, t_i, s_i)}{\sum_s C(w_{i-1}, t_i, s)} = \frac{C(w_{i-1}, t_i, s_i)}{C(w_{i-1}, t_i)} \quad (5)$$

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 (4) can be computed similarly. If a large training corpus is available, the parameters in equation (4) can be easily estimated using the MLE technique. To account for zero probabilities for any cases that are not observed in the training data, 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

$$\begin{aligned} P'(s_i \mid w_{i-1}, t_i) &= \lambda P(s_i \mid w_{i-1}, t_i) + (1-\lambda) P(s_i \mid t_i) \\ P'(w_i \mid w_{i-1}, s_i, t_i) &= \begin{pmatrix} \beta P(w_i \mid w_{i-1}, s_i, t_i) + \\ (1-\beta) P(w_i \mid s_i, t_i) \end{pmatrix} \\ P'(t_i \mid w_{i-1}, t_{i-1}) &= \alpha P(t_i \mid w_{i-1}, t_{i-1}) + (1-\alpha) P(t_i \mid t_{i-1}) \end{aligned} \quad (6)$$

Where λ, β and α denote the interpolation coefficients.

3.3 Information Propagation using Entity's Synonyms, Antonyms and Related Words

To cover more language phenomenon and large domains, in the training step, we have employed a new technique to automatically propagate information of each expert tagged entity to its synonyms, antonyms, similar words and related words. Figure 2 illustrates an example. As mentioned above, an entity can be a single word or a phrase. By expanding each single word to a list of its related words, different word combinations can be formed. In Figure 2, the sentence *"Good picture quality"* is an expert tagged opinion sentence. During the training course, the system looks up synonyms and antonyms for opinion entities. The tag of the original opinion entity "*good*", <OPINION_POS_EXP> (*positive opinion*), gets propagated to each synonym of "*good*" (red box on the left in Figure 2). The negative tag <OPINION_NEG_EXP> gets propagated to "*good*"'s antonyms (dark red box on the bottom left). Similarly, for each single word in other entity types, similar words and related words are looked up. The tag of the original word gets propagated to each newly discovered related word (blue boxes). Using this expansion, a number of bi-gram combinations (green arrows) can be obtained. In this example, there are several possible instances derived from *"Good picture quality"*, such as "Decent picture quality", "Poor image quality", and etc.

Obviously, only *"Good picture quality"* is the expert tagged truth data. All other combinations generated from expansion might contain noise. To reduce the noise impact, a confidence weight is given to each bi-gram combination when computing the MLE values in equation 4. We empirically set $W_1 = 1$ and $W_2 = 0.01$ for expert tagged combinations and combinations obtained from expansion, respectively.

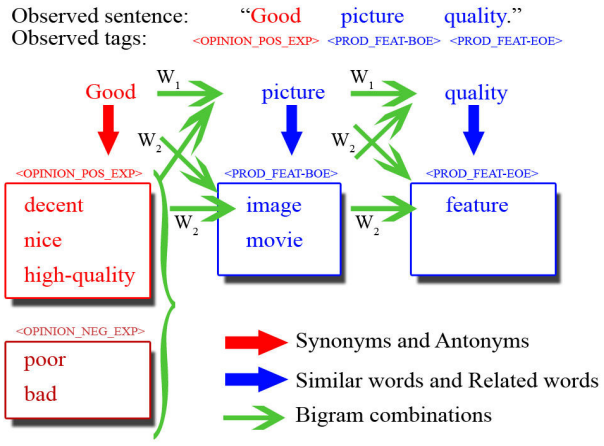

Figure 2. Information propagation using entity's synonyms, antonyms, similar words and related words

A dictionary program has been built to return an input word's synonyms, antonyms, similar words and related words using Microsoft Word's thesaurus. The reason we decided not to use WordNet [1] for this purpose is that after experimenting with WordNet, we found it returned too many less commonly used synonyms and antonyms. However, most reviewers are prone to use commonly used words to express their opinions. Expanding entities using less frequently used terms creates more noise and affects the classifier's performance.

3.4 Token Transformations

Another problem with many entities is that they may be overly specific. For example, *"I love its 28mm lens"* and "I love its *300mm* lens". Both sentences talk about lens. They could be ideally grouped together as *"I love its Xmm lens"* where *X* can be any numerical value. This transformation generalizes the information contained in sentences and is useful in solving the problem of sparseness in the training data. In our framework, we use this transformation to handle high detailed information in sentences such as *Model Number*, *Focal Length*, and *ISO* (the sensitivity of image sensor). The following three transformations are performed. Given the transformation table 4:

Table 4. The transformation table

	Regular Expression	Examples
1	^|[`";,!?:()\[\]]|\.+$\|'$	Match ' ` " ; , ! ? : ( ) [ ] .
2	\d+|\d+\.\d+|\d+\-\d+	Match 3, 3.5, 3-4
3	[A-Z-]+\d+([A-Za-z]+)?	Match DMC-LS70S, P5100, A570IS

1. Remove any punctuation that matches regular expression 1.
2. Transform any token that matches regular expression 2 but does not match regular expression 3 to symbol "#NUM#".
3. Transform any token that matches regular expression 3 to symbol "#MODEL#".

[1] http://wordnet.princeton.edu/

Step 1 removes any unnecessary punctuations in a token. Step 2 generalizes all numerical expressions except model numbers. For the previous example, both opinion sentences will be transformed into *"I love its #NUM#mm lens"*. Step 3 generalizes product model numbers. This transformation step was applied to both the training and classification.

3.5 Decoding

Based on the above model, the decoding algorithm aims at finding the most probable sequence of hybrid tags for a given sequence of known words and corresponding parts of speech. As discussed above, a hybrid tag of an observable word involves a category tag and a pattern tag. The candidate hybrid tags of a known word are a combination of its candidate category tags and its candidate pattern tags. The Viterbi algorithm is employed to score all candidate hybrid tags with the proposed language models, and then search the best path that has the maximal score.

3.6 Opinion Sentence Extraction

This step identifies opinion sentences in the reviews. Opinion sentences in our work are defined as sentences that express opinions on product related entities. In our system, 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.

3.7 Determining Opinion Orientation

The following step further classifies opinion orientation given each identified product entity. Due to the complexity and flexibility 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, for each recognized product entity, we first search its matching opinion entity, which is defined as the nearest opinion entity 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 sentence context are employed to address specific language constructs, such as the presence of negation words (e.g., not), which may change the opinion orientation.

Specifically, we check the presence of any negation words (e.g., not, didn't, don't) within five-word distance in front of an opinion entity and changes opinion orientation accordingly, except

1. A negation word appears in front of a coordinating conjunction (e.g. and, or, but).
2. A negation word appears after the appearance of a product entity during the backward search within the five-word window.
3. A negation word appears before another negation word.

The coordinating conjunctions such as "*but*" and prepositions such as "*except*" and "apart from" are addressed as follows: if opinion entity is in front of the corresponding product entity and prepositions such as "but/except" appear between opinion entity and product entity, then the opinion orientation for the corresponding product entity is updated with the opposite of its initial orientation.

4. EXPERIMENTS

We used Amazon's digital camera reviews as the evaluation dataset. The reviews for the first 16 unique cameras listed on Amazon 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 Stanford NLP Group[2] and default settings were used.

4.1 Training Design

After downloading and pre-processing, there were altogether 1728 review documents obtained. We separated the documents into 2 sets. One set (293 documents for 6 cameras) was manually tagged. Opinion sentences were identified and 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 (described next) to self-learn new vocabularies.

4.2 Bootstrapping

Labeling training documents manually is a labor intensive task. Thus, it would be nice if the system can identify 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. 3 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.
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. Each worker first 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

[2] http://nlp.stanford.edu/software/lex-parser.shtml

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 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.

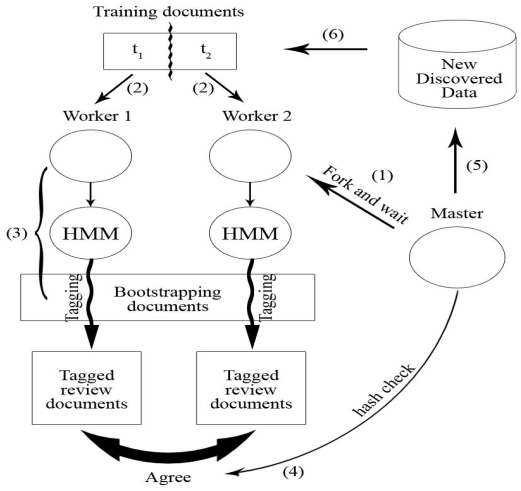

Figure 3. The Bootstrapping Process

One characteristic of this bootstrap process is each HMM classifier always has its set of unique training data. Additionally, in each bootstrap cycle, both HMM classifiers' training data are different from the previous cycle.

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 (and high confidence) data collected from the original training set and bootstrap data set, and was then applied to our test data and evaluated against the baseline.

The effectiveness of the proposed framework was evaluated by measuring 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 as one of four pre-defined entity types. 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 type are identified with correct orientation (positive or negative).

4.3.1 Baseline System

We have designed and implemented a rule-based baseline system motivated by [3] and [9]'s approaches. [3] describes 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 5 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 [9], 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 words were 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.

Table 5. 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)

4.3.2 Further Experiments on Hu's Corpus

In addition to using the dataset downloaded from Amazon.com, the publicly available Hu and Liu's corpus [9] was also used as evaluation data. Their corpus is for product review summarization, which is closely related to our work. However, there are two major differences between Hu's task and ours. 1) Hu's work is focused on summarization where extracting

generic terms and frequent terms are their major concern, whereas our work is focused on extracting high detailed product entities and both frequent and infrequent entities are considered equally important; 2) Other than identifying desired entities, we further classify these entities into different categories. This could lead to the automatic construction of a hierarchical relationship (such as the Entity-Relationship schema) from free texts between product entities and their associated attributes. Due to these differences, some extra work was done on Hu's corpus. First, Hu's corpus did not include any entity type information. We manually labeled the desired entity types so that the evaluation program can measure the system performance for each entity type. Second, if reviewers use specific terms (e.g., optical viewfinder) instead of generic terms (e.g., viewfinder), specific terms are considered as unique correct terms. In other words, identifying entities capturing finest details of the product is one of our aims. The following example illustrates the major difference between Hu's labeled data and ours.

"The menus are easy to navigate and the buttons are easy to use."

Hu's labels: [menu|+] [button|+]

Our labels: [menu|+ [navigate|+]] [buttons|+ [use|+]]

Each bracket represents an entity (entity type is not shown). The '+' symbol represents positive polarity. In our corpus, "*navigate*" is considered as a feature of "*menu*" and "*use*"(usability) as a feature of "*buttons*". "*Menus*" and "*buttons*" are labeled as *product function* and *product component*, respectively.

4.4 Evaluation Results and Discussions

The detailed evaluation results are presented in Table 6 and Table 7. As a post analysis, the proposed machine learning framework 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 observed our approach effectively identified highly specific product entities and opinion expressions (usually complex phrases) and self-learned new vocabularies based on the patterns it has seen from the training data (the examples are shown in Table 8).

Another observation is in addition to effectively extracting frequent entities, the system also excels in identifying important but infrequently mentioned entities, which was under-analyzed or ignored by previously proposed methods. In this work, we propose infrequent entities, such as "*on/off button*", "*battery/memory compartment*" and "*focus assist light*" (identified by the system but only occur once or twice in the dataset), could be useful product descriptors when answering user's specific queries in many web applications (e.g., recommender systems). In such applications, frequent generic features might be satisfied by most candidate products. However, infrequent product-specific features might be better able to differentiate different products (e.g. recommending a list of cameras which have positive feedbacks on "kids mode"). Additionally, the user's preferences could be highly specific. For example, "*automatic white balance*", "*custom white balance*" and "*preset white balance*" represent different user preferences and a recommender system should be able to distinguish among these to answer the user's specific queries. Such information can be effectively extracted by the proposed learning system.

In this paper, we also propose the potential non-noun product entities, such as "engineered" and "operated" (an example is shown below). These non-noun entities were ignored by previously proposed approaches which were based on the assumption that product entities must be noun or noun phrases. Our system can well identify these overlooked product entity information.

Operated = 2 ("= X" represents the number of occurrences of an entity)

The <PROD_FUNCTION> zoom </PROD_FUNCTION> is <OPINION_POS_EXP> easily </OPINION_POS_EXP> <PROD_FEAT> operated </PROD_FEAT> without looking.

The/DT zoom/NN is/VBZ easily/RB operated/VBN without/IN looking/VBG

4.5 User Interface

Figure 4 shows the *OpinionMiner* system interface and the format of answers we would like to provide for the user. In this interface, opinion sentences are identified; product related entities and opinion related entities appearing in opinion sentences are recognized and highlighted using different colors (corresponding to different entity types such as <PROD_FEAT>, <PROD_FUNCTION> and <OPINION_POS_EXP>).

5. CONCLUSIONS

In this paper, a novel and robust machine learning system is designed for opinion mining and extraction. The model provides solutions for several problems that have not been addressed by previous approaches. Specifically,

- The model naturally integrates multiple linguistic features (e.g., part-of-speech, phrases' internal formation patterns, surrounding contextual clues) into automatic learning.
- The system can predict new potential product and opinion entities 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. This capability 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 bootstrapping approach combining active learning through committee votes and L-HMM is employed to handle situations in which collecting a large training set could be expensive and difficult to accomplish.

The existing problems are:

(1) People like to describe a long story about their experiences. For example, some people like to describe how bad/good their former cameras were. This influences the system performance

on some camera reviews in the experiments. We are looking into this issue further.

(2) Some researchers suggested pronoun resolution. We have applied pronoun resolution to each sentence in our experiments. However, the results are not satisfying. We found pronoun resolution caused too many false positives. After a closer look at the data, we consider sentence classification might be needed to determine which sentences should perform pronoun resolution. This is also left for future work.

6. REFERENCES

[1] Lee, S. Z., Tsujii, J., and Rim, H. C. 2000. Lexicalized Hidden Markov Models for Part-of-Speech Tagging. In Proceedings of the 18th International Conference on Computational Linguistics (COLING'00), 481-487.

[2] Fu, G. and Luke, K. K. 2005. Chinese Named Entity Recognition using Lexicalized HMMs. ACM SIGKDD Explorations Newsletter 7,1 (2005), 19-25.

[3] Turney, P. D. 2002. Thumbs up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02), 417-424.

[4] Turney, P. D. and Littman, M. L. 2003. Measuring praise and criticism: Inference of semantic orientation from association. ACM Trans. On Information Systems, 21, 4 (2003), 315-346.

[5] Dave, K., Lawrence, S., and Pennock, D. M. 2003. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. In Proceedings of the 12th international conference on World Wide Web (WWW'03), 519-528.

[6] Pang, B., Lee, L., and Vaithyanathan, S. 2002. Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP'02), 79-86.

[7] Pang, B. and Lee, L. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics (ACL'04), 271-278.

[8] Das, S. and Chen, M. 2001. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In Proceedings of the 8th Asia Pacific Finance Association Annual Conference (APFA'01).

[9] Hu, M. and Liu, B. 2004. Mining and Summarizing Customer Reviews. In Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04), 168-177

[10] Zhuang, L., Jing, F., and Zhu, X. 2006. Movie Review Mining and Summarization. In Proceedings of the International Conference on Information and Knowledge Management (CIKM'06), 43-50.

[11] Popescu, A. and Etzioni, O. 2005. Extracting Product Features and Opinions from Reviews. In Proceeding of 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP'05), 339-346.

[12] Ding, X., Liu, B., and Yu, P. S. 2008. A Holistic Lexicon-based Approach to Opinion Mining. In Proceeding of the international conference on Web Search and Web Data Mining (WSDM'08), 231-239.

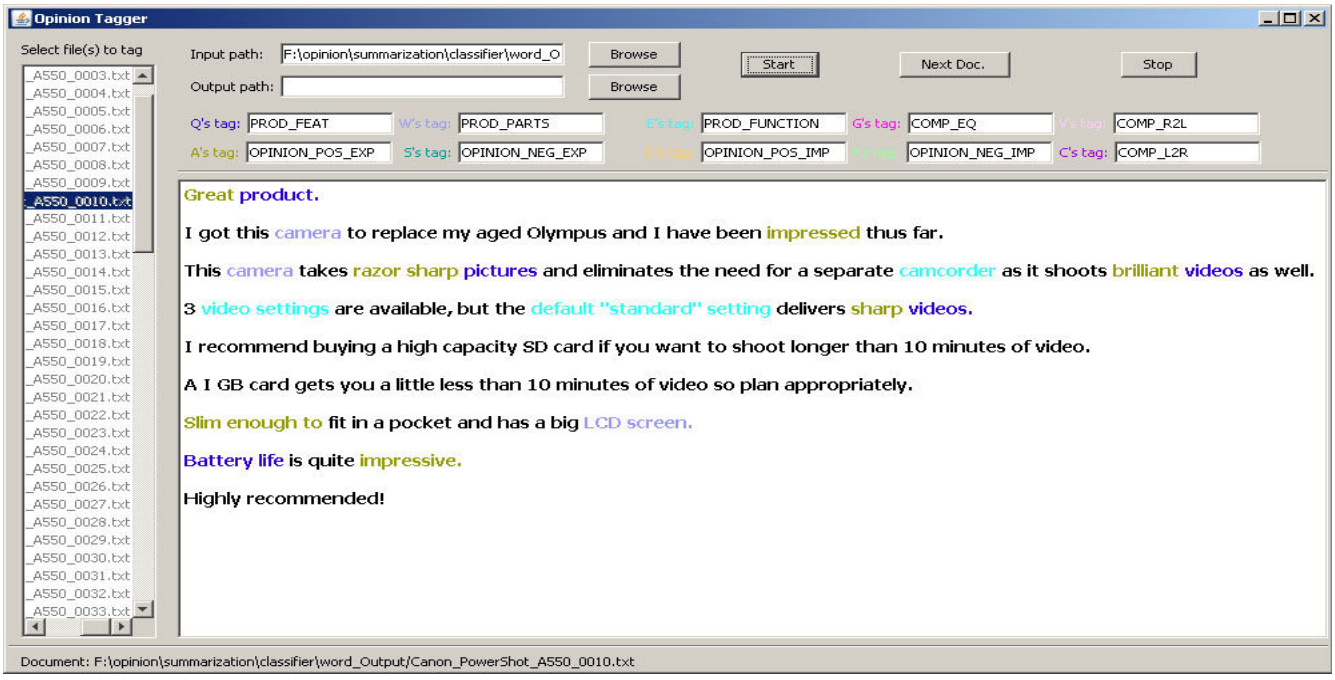

Figure 4. The user interface and example output (Opinion sentences are identified and product entities and opinion entities appearing in opinion sentences are recognized and highlighted using different colors (corresponding to different entity types))

Table 6. Experimental results on entity extraction (R: Recall; P: Precision; F: F-score; VE: Vocabulary Expansion; BS: Bootstrapping)

Products	Methods	Feature Entities (%)			Component Entities (%)			Function Entities (%)			All Entities (%)		
		R	P	F	R	P	F	R	P	F	R	P	F
Camera A	L-HMM+POS+VE+BS	**86.5**	**79.9**	**83.1**	**82.7**	**81.7**	**82.2**	**65.6**	**79.3**	**71.8**	**82.7**	**80.3**	**81.5**
	L-HMM+POS+VE	82.4	75.8	78.9	74.4	72.5	73.4	65.6	67.7	66.7	77.8	73.9	75.8
	L-HMM+POS	82.0	77.7	79.8	73.1	73.1	73.1	65.6	70.0	67.7	77.2	75.4	76.3
	L-HMM	80.7	75.9	78.2	70.4	70.3	70.3	60.2	67.2	63.5	75.8	73.2	74.5
	Baseline	-	-	-	-	-	-	-	-	-	20.4	30.0	24.3
Camera B	L-HMM+POS+VE+BS	**88.4**	**73.5**	**80.3**	**80.9**	**73.8**	**77.2**	**65.2**	**88.2**	**75.0**	**83.7**	**74.4**	**78.8**
	L-HMM+POS+VE	86.4	69.8	77.2	79.8	75.0	77.3	56.5	76.5	65.0	81.4	71.9	76.4
	L-HMM+POS	84.4	72.1	77.7	74.5	70.9	72.7	52.2	75.0	61.5	78.0	71.8	74.8
	L-HMM	80.7	71.5	75.8	71.7	70.5	71.1	47.8	64.7	55.0	74.8	70.9	72.8
	Baseline	-	-	-	-	-	-	-	-	-	15.5	24.3	18.9
Camera C	L-HMM+POS+VE+BS	**80.5**	**79.8**	**80.2**	**97.6**	**79.4**	**87.6**	**72.7**	**82.7**	**77.4**	**85.3**	**80.3**	**82.7**
	L-HMM+POS+VE	75.3	77.3	76.3	97.6	76.9	86.0	72.7	78.9	75.6	82.2	77.5	79.8
	L-HMM+POS	74.0	80.3	77.0	97.6	78.4	86.9	63.6	80.5	71.1	80.6	79.6	80.1
	L-HMM	70.3	77.3	73.7	97.6	74.1	84.2	63.6	70.0	66.7	78.2	75.5	76.9
	Baseline	-	-	-	-	-	-	-	-	-	17.1	23.7	19.8
Hu's corpus Camera D	L-HMM+POS+VE+BS	**88.6**	**62.0**	**72.9**	**82.2**	**70.2**	**75.7**	**86.9**	**76.8**	**81.5**	**86.0**	**66.7**	**75.1**

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

Products	Methods	Opinion sentence extraction (sentence level)				Entity-opinion pair orientation (feature level)		
		R(%)	P(%)	F(%)		R(%)	P(%)	F(%)
Camera A	L-HMM+POS+VE+BS	**91.41**	**85.81**	**88.52**		**78.59**	**75.76**	**77.15**
	L-HMM+POS+VE	89.00	82.48	85.62		74.46	70.45	72.40
	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+VE+BS	**89.16**	**80.87**	**84.81**		**75.00**	**66.67**	**70.59**
	L-HMM+POS+VE	89.76	80.54	84.90		71.97	63.33	67.38
	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	51.16		13.26	20.71	16.17
Camera C	L-HMM+POS+VE+BS	**87.36**	**85.85**	**86.60**		**79.84**	**73.55**	**76.57**
	L-HMM+POS+VE	82.76	82.76	82.76		77.52	73.53	75.47
	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
Hu's corpus camera D	L-HMM+POS+VE+BS	**85.58**	**69.17**	**76.50**		**73.32**	**60.03**	**66.01**

Table 8. Examples of self-learned vocabularies

Auto = 3 auto setting = 2 auto flash* = 3 auto focus = 1 auto function* = 1 auto ISO setting* = 1	auto red eye correction = 1 auto stabilizer* = 2 auto white balance* = 3 automatic = 2 automatic setting = 5 automatic fill-flash* = 1	automatic focus* = 1 automatic white balance = 1 automatic zoom* = 1 automatic functions* = 1 automatic point-and-shoot mode* = 1
* represents a new self-learned vocabulary		"= X" represents the number of occurrences of an entity