A Maximum Entropy Model for Product Feature Extraction in Online Customer Reviews

Gamgarn Somprasertsri
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok 10520, Thailand
s7066001@kmitl.ac.th

Pattarachai Lalitrojwong
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok 10520, Thailand
pattarachai@it.kmitl.ac.th

***Abstract*—Product feature extraction is an important task of review mining and summarization. The task of product feature extraction is to find product features that customers refer to in their topic reviews. It would be useful to characterize the opinions which they review or express about the products. In this paper, we propose an approach to product feature extraction using a maximum entropy model. Maximum entropy is a probability distribution estimation technique. It is widely used for classification problems in natural language processing, such as question answering, information extraction, and part-of-speech tagging. The underlying principle of maximum entropy is that without external knowledge, one should prefer distributions that are uniform. Using a maximum entropy approach, at first we extract features from the corpus, train maximum entropy model with an annotated corpus, and then use it with additional product feature discovery to extract product features from customer reviews. Our experimental results show that this approach can work effectively for product feature extraction with 71.88% precision and 75.23% recall.**

***Keywords*—product feature extraction, maximum entropy model, text mining, review mining and summarization**

I. Introduction

Recently, the number of online shopping customers has been increased due to the rapid growth of e-commerce, an increasing number of online merchants, and more and more people becoming comfortable with the internet. To enhance customer satisfaction, merchants and product manufacturers allow customers to review or express opinions on the products they buy from their websites, for instance, amazon.com, cnet.com, and epinions.com. Online customer reviews become the source of information which is very useful to both potential customers and product manufacturers. People read them for making a decision on whether to purchase the product. For a product manufacturer, knowing the preferences of customers is highly valuable for product development, marketing and consumer relationship management. Mining reviews mostly in free-form text can be extremely expensive. Besides, it is hard to find useful information from plenty of customer reviews. This trend has raised many interesting and challenging research topics such as subjectivity classification, sentiment classification, and review mining and summarization.

Subjectivity classification is distinguishing sentences or documents that present opinions from factual information, as in [1][2]. Many natural language processing applications could benefit from being able to distinguish between factual and subjective information such as question answering, information extraction, and so on. The task of sentiment classification is to judge whether a review expresses a positive or negative opinion. For example, [3][4] developed methods for document level sentiment classification. The systems assign a positive or negative sentiment for the whole review document. Sentiment of phrases and sentences has also been studied in [5][6]. Even if sentiment classification is useful, it does not find what the reviewer liked and disliked. Review mining and summarization is the task of producing a sentiment summary, which consists of sentences from reviews that capture the author's opinion. Review summarization is only interested in features or objects on which customers have opinions. It also determines whether the opinions are positive or negative. This makes it differ from traditional text summarization. Most existing works on review mining and summarization mainly focus on product reviews. For example, [7][8][9] concentrated on mining and summarizing reviews by extracting opinion sentences regarding product features. In another domain, [10] proposed a multi-knowledge based approach for movie review mining and summarization.

In general, mining and summarizing customer reviews involve three tasks (Figure 1): firstly, feature extraction identifies and extracts object features that have been commented in each review; secondly, sentiment assignment determines the polarity of each feature to be positive or negative; and thirdly, summary visualization summarizes the result in order to show this result more effectively. Product feature extraction is an important task of review mining and summarization. Identifying product features would be useful to characterize the opinions the customers review or express about the products. In this paper, we only focus on the first task of review mining and summarization. We propose a maximum entropy model for extracting product features. Our goal is to investigate whether the maximum entropy model is suitable for automatic product feature extraction. Our experimental results show that this approach is effective.

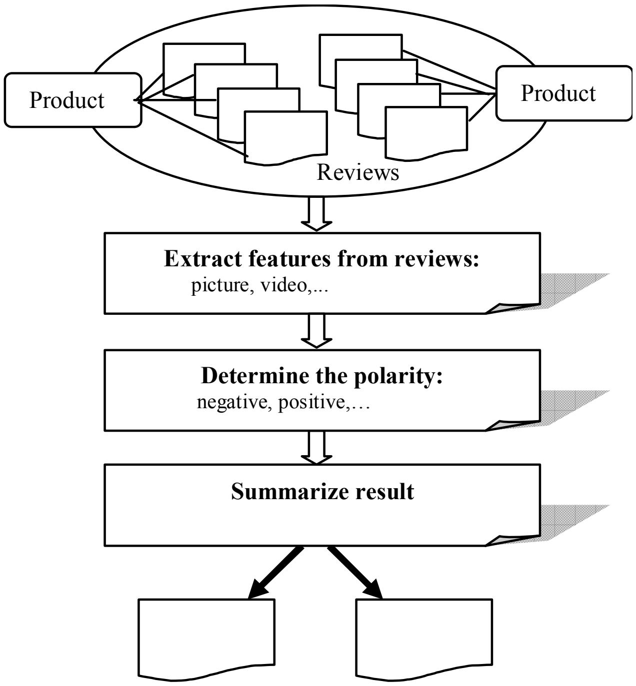

Figure 1. Review mining and summarization process

The rest of this paper is organized as follows. Section II describes related work on the task of product feature extraction. Section III introduces the maximum entropy model. Section IV discusses how to extract product features from online customer reviews using maximum entropy. Section V presents and discusses experimental results. Finally, Section VI concludes our work.

II. RELATED WORK

Hu and Liu's work in [11] can be considered as the pioneer work on feature extraction from reviews. Their feature extraction algorithm is based on heuristics that depend on feature terms' respective occurrence counts. They use association rule mining based on the Apriori algorithm to extract frequent itemsets as explicit product features (only in the form of noun phrases). In association rule mining, the algorithm does not consider the position of the words in a sentence. In order to remove incorrect frequent features, they use feature pruning that consists of compactness pruning and redundancy pruning. To improve the work over [11], Liu, Hu, and Cheng [12] propose a technique based on language pattern mining to identify product features from pros and cons in reviews in the form of short sentences. They also make an effort to extract implicit features.

Popescu and Etzioni [8] developed an unsupervised information extraction system called OPINE extracting product features and opinions from reviews. OPINE first extracts noun phrases from reviews and retains those with frequency greater than an experimentally set threshold and then assesses those by OPINE's feature assessor for extracting explicit features. The assessor evaluates a noun phrase by computing a Point-wise Mutual Information score between the phrase and meronymy discriminators associated with the product class.

Carenini, Ng, and Zwart [13] proposed feature extraction for capturing knowledge from product reviews. In their method, the output of Hu and Liu's system [11] was used as the input to their system, and map the input to the user-defined taxonomy features hierarchy thereby eliminating redundancy and providing conceptual organization.

Finally, Yi and Niblack [14] developed a set of feature term extraction heuristics and selection algorithms for extracting a feature term from product reviews. The feature term is a part of relationship with the given topic, an attribute of relationship with the given topic, and an attribute of relationship with a known feature of the given topic. In the first step, they extract a noun phrase with the Beginning define Base Noun Phrase (bBNP) heuristics. Then, they select a feature term from the noun phrase using the likelihood score.

Our motivation for building the product feature extraction system described in this paper is that the grammars and word information may be useful for determining whether the word is a product feature or non-product feature. For this motivation, we will employ the maximum entropy model which can indeed combine various features of grammars or words into a probability model for product feature extraction in online customer reviews.

III. MAXIMUM ENTROPY MODEL

The maximum entropy (ME) model was first described by Jaynes in [15] and more recently in a draft manuscript available on the Web [16]. The maximum entropy model is a framework for integrating information from many heterogeneous information sources for classification [17]. This model implements the intuition that the best model is the one consistent with the set of constraints imposed by the evidence but otherwise is as uniform as possible [18]. The maximum entropy approach has in recent years been used for a wide variety of classification problems in natural language processing, such as sentence boundary detection [19][20], information extraction [21], and part-of-speech tagging [22]. The underlying principle of maximum entropy is that without external knowledge, one should prefer distributions that are uniform. For our work, we use maximum entropy model to extract product features from online customer reviews. This task can be re-formulated as a classification problem, in which the task is to observe some linguistic context $x \in X$ and predict the correct linguistic class $y \in Y$.

We can implement classifier cl: $X \rightarrow Y$ with a conditional probability model by simply choosing the class y with the highest conditional probability in the context x:

$$cl(x) = \arg\max_{y} p(y \mid x) \tag{1}$$

The conditional probability $p(y|x)$ is defined as follows [20]:

$$p(y \mid x) = \frac{1}{Z(x)} \prod_{i=1}^{k} \alpha_i^{f_i(x,y)} \quad (2)$$

$$Z(x) = \sum_y \prod_i \alpha_i^{f_i(x,y)} \quad (3)$$

where y refers to the outcome, x is the history (or context), k is the number of features and $Z(x)$ is a normalization factor to ensure that $\sum_y p(y \mid x) = 1$. Each parameter α_i , corresponds to one feature f_i and can be interpreted as a weight for that feature. The parameters α_i are estimated by a procedure called Generalized Iterative Scaling (GIS) [23]. This is an iterative method that improves the estimation of the parameters at each iteration.

Under the maximum entropy framework, the probability for a class y and object x depends solely on the features that are active for the pair (x, y), where a feature is defined here as a function f: X × Y → {0, 1} that maps a pair (x, y) to either 0 or 1. The feature is defined as follows:

$$f_{y'}(x, y) = \begin{cases} 1 & if\ y' = y\ and\ cp(x) = true \\ 0 & otherwise \end{cases} \quad (4)$$

where $cp(x)$ is contextual predication that returns true or false, corresponding to the presence or absence of useful information in some context, or history $x \in$ X. For example, to predict which the class of target word belongs (as shown in Table I). The classifier considers surrounding context of the target word. If the target word is product feature and the previous word is "the", a feature function can be set as Equation 5. On the other hand, if the target word is non-product feature and it contains small letters, a feature function can be set as Equation 6.

$$f_i(x_j, y_j) = \begin{cases} 1 & if\ y_j = YES\ and\ previousword = "the" \\ 0 & otherwise \end{cases} \quad (5)$$

$$f_i(x_j, y_j) = \begin{cases} 1 & if\ y_j = NO\ and\ capital(word_j) = False \\ 0 & otherwise \end{cases} \quad (6)$$

TABLE I. CLASSES DEFINED FOR THE CLASSIFICATION TASK

Class	Description
YES	Word claimed to be product feature
NO	Word claimed to be non-product feature

IV. PRODUCT FEATURE EXTRACTION

The main objective of this research is to identify product features from reviews, namely, product feature extraction. The product feature can be a brand name, a model name of a commodity, a property, a part, a feature of a product, a related concept, and a part of related concept [8]. In general, such product features are often the product itself or its specific features, such as *quality* (e.g. "*The picture quality is amazing*"). The product features mentioned in a review can be classified into two types: explicit and implicit features. For example, the sentence from a review of a DVD player, *"the sound is good"* shows the explicit product feature. It means that the customer is satisfied with the *sound* of the DVD player, and the *sound* is explicitly mentioned in the sentence. On the contrary, the sentence *"it does tend to run quite hot"* shows that the customer is talking about the *heat* of the DVD player, but the *heat* is not explicitly mentioned in the sentence. Therefore, *heat* is an implicit feature in this sentence. Extracting features from free-form text of a review is a challenging task because of the use of natural language.

A. The Overview of System

In this research work, we aim to extract explicit product features commented by customers. We leave finding implicit features to our future work. We define the product feature extraction problem as a classification task: given a sequence of words ($x_1, x_2, \ldots, x_n$) in a sentence, we generate a sequence of labels ($y_1, y_2, \ldots, y_n$) indicating whether the word is a product feature or non-product feature. We apply the maximum entropy model to extract the product features. The processes involve two phases: firstly to train the ME model; and secondly to test the model extracting product features from unlabeled reviews. We first prepare a training data set by manually labeling product features of reviews. The process of product feature extraction is shown in Figure 2 which can be described as follows:

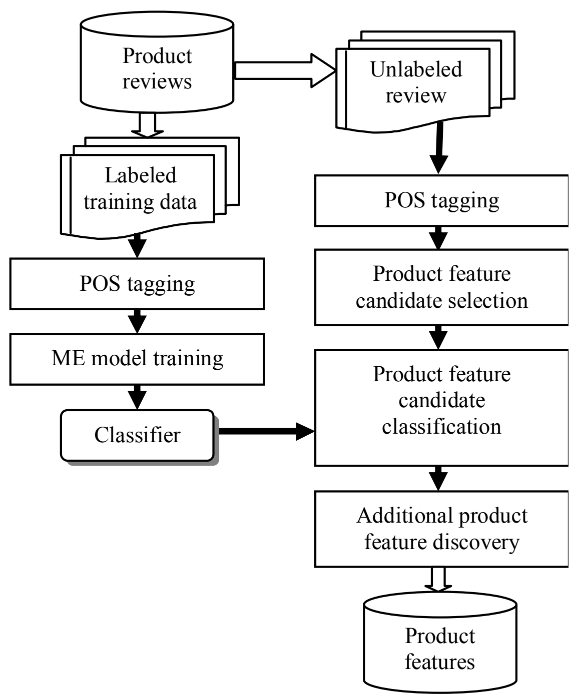

Figure 2. Process of product feature extraction

1) Part-of-speech tagging

We utilize the Stanford lexicalized parser [24] to analyze each review and tag the part of speech of each word. The following is an example of the output such as nouns, verbs, adjectives and so on. *The/DT macro/NN mode/NN is/VBZ* *exceptional/JJ ,/, the/DT pictures/NNS are/VBP very/RB clear/JJ and/CC you/PRP can/MD take/VB the/DT pictures/NNS with/IN the/DT lens/NN unbelievably/NN close/RB the/DT subject/NN.*

2) ME model training

Training the ME model involves two steps. The first step of the maximum entropy approach is to extract features or important information in order to constrain the model accordingly. For product feature extraction, the features or important information can be the context information, the part of speech information and so on which will be explored in the next section. The second step is to train a model by using these features according to Equation 2.

3) Product feature candidate selection

For product feature extraction, the first step is POS tagging. After tagging the sentence, the next step is identifying product feature candidates. This step selects words from a tagged sentence such as nouns and adjectives, which are indicated by NN and JJ respectively. Using only nouns and adjectives is reasonable because most product features are nouns; however, some adjectives may appear as product features.

4) Product feature candidate classification

The trained model is employed to classify product feature candidates from unlabeled reviews after POS tagging. We will simply choose the class with the highest conditional probability *p* according to Equation 1.

5) Additional product feature discovery

This step aims to improve the performance of product feature extraction. After extracting product features by the ME based classifier, we applied a natural language processing technique to deal with compound product feature candidates which are not extracted by the classifier. The product feature candidates will be extracted if they or their head noun matches the product features extracted by the classifier. We take the sentence "*The scroll wheel was a nice idea to keep less clutter*" as an example. The word "*wheel*" is the head noun of "*scroll wheel*" extracted by the classifier, thus the "*scroll*" will be extracted as a product feature. The pseudo code of this process is provided in Figure 3.

```
Procedure ExtractRefinement(feature_list, review)
begin
 for each sentence in review
   for each candidate w_i in sentence
   begin
     if (w_i match f_j in feature_list) and
          (w_i's ME score < threshold) then
        w_i's class = product feature
     else if (w_{i+1}'s POS tag is noun) and
              (w_{i+1}'s class is product feature) then
            w_i's class = product feature
   endfor;
 endfor;
end
```

Figure 3. Pseudo code of additional product feature discovery

B. Features forProduct Feature Classification

To use the maximum entropy to extract product features, we define features or important information in order to constrain the model. We denote the features employed for learning as learning features, discriminative from the product features we discussed above. We compute several features automatically. Table II summarizes the features we used for our model and the symbols we will use in the rest of this paper. The features can be described as follows.

TABLE II. FEATURES IN OUR MODEL

Symbol	Feature name	Description
F1	Context	words in a [-4, +4] window centered on w_i
F2	Part-of-Speech Tag	POS tags in a [-4, +4] window centered on t_i
F3	Rare Word	words which occur less than five times in the training set
F4	Alphanumeric	words containing letters and numerals
F5	Capitalized	words starting with a capital letter

1) Context

Words preceding or following the target word may be useful for determining its category. For example, "*the sound is wonderful*". If the target word is "*sound*" and the following words are "*is*" and "*wonderful*", then this will help the model to classify "*sound*" as a product feature. The more context words analyzed, the better and more precise the results. However, widening the context window quickly leads to an explosion of the number of possibilities to calculate. In our experiment, a suitable window size is +/-4 words.

2) Part-of-Speech Tag

Part of speech tag is quite useful for identifying product features. Verbs and prepositions usually indicate the product feature boundaries whereas nouns and adjectives are usually good candidates for product features.

3) Rare Word

Rare word information may be useful for identifying product features. It is common that a customer review contains many things that are not directly related to product features. Different customers usually have different stories. Those frequent noun-noun phrases (non-rare words) are likely to be product features and infrequent noun-noun phrases (rare words) are likely to be non-product features. A rare word in our work denotes a word which occurs less than five times in the training set. The count of five was chosen by subjective inspection of words in the training data.

4) Orthography

Orthography is a characteristic of words such as words containing letters and numerals, words starting with a capital letter, and so on. This information may be useful for determining whether the word is a product feature or non-product feature. Two types of orthography considered in our model are alphanumeric and capitalized. Alphanumeric means words containing letters and numerals. Capitalized means words starting with a capital letter. The regular expression of the alphanumeric and the capitalized are presented in Table III.

TABLE III. ORTHOGRAPHY FOR OUR MODEL

Feature name	Regular Expression
Alphanumeric	.*[A-Za-z].*[0-9].*| .*[0-9].*[A-Za-z].*
Capitalized	[A-Z][a-z]+

V. EXPERIMENTS

For our experiments, we used reviews on electronic products such as digital cameras and MP3 players from the Amazon web site. We annotated a randomly selected sample of 1,555 sentences for product feature extraction. Each word of each sentence was classified as a product feature or a non-product feature. This set of data was split into a training set of 1,255 sentences and a testing set of 300 sentences. Words in sentences are represented as vectors of binary features. The training originally yielded 13,066 features. We have used Maxent toolkit version 2.4.0 from [25]. The model was trained with features in the form of words in sentence contexts, POS tags, Orthography and rare word. The parameters of the maximum entropy model can be trained with 100 iterations of the Generalized Iterative Scaling algorithm which must be calculated repeatedly. Normally, it is a good "rule of thumb" to carry out 100 iterations [20]. More iteration would not increase the accuracy of the parameters.

The evaluation methods are precision (P), recall (R), and F-score (F). They are defined as follow:

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$F = \frac{2\,PR}{P + R}$$

where TP means the number of product features extracted correctly; FP means the number of words mistakenly claimed to be product features; and FN means the number of product features not extracted.

We conducted these experiments with three goals: firstly, to investigate how well our product feature extraction model with different window sizes of the contexts and POS tags; secondly, to investigate how well the model with different feature combinations performs on the customer reviews; and thirdly, to see how well our approach (ME model with additional product feature discovery) performs the product feature extraction compare to baseline (ME model without additional product feature discovery).

TABLE IV. PERFORMANCE OF THE MODEL USING CONTEXT WITH DIFFERENT WINDOW SIZES

Window Size	Precision (%)	Recall (%)	F-Score (%)
+/-1 word	**72.18**	61.21	66.25
+/-2 words	71.58	61.21	65.99
+/-3 words	69.66	61.68	65.43
+/-4 words	71.20	**62.38**	**66.50**

TABLE V. PERFORMANCE OF THE MODEL USING PART-OF-SPEECH TAG WITH DIFFERENT WINDOW SIZES

Window Size	Precision (%)	Recall (%)	F-Score (%)
+/-1 word	69.92	58.64	63.79
+/-2 words	72.38	58.18	64.51
+/-3 words	72.14	57.48	63.98
+/-4 words	**73.73**	**60.98**	**66.75**

TABLE VI. PERFORMANCE OF THE MODEL USING THE COMBINATION OF FEATURES

Features	Precision (%)	Recall (%)	F-Score (%)
F1F2	76.80	57.24	65.60
F1F2F3	77.74	**57.94**	**66.40**
F1F2F4	77.53	57.24	65.86
F1F2F5	77.19	57.71	66.04
F1F2F3F4	**78.10**	57.48	66.22
F1F2F3F5	77.85	57.48	66.13
F1F2F4F5	76.88	57.48	65.78
F1F2F3F4F5	77.64	56.78	65.59

TABLE VII. THE PERFORMANCE OF SYSTEM FOR PRODUCT FEATURE EXTRACTION

	Precision (%)	Recall (%)	F-Score (%)
Baseline	**78.10**	57.48	66.22
Our approach	71.88	**75.23**	**73.52**

Tables IV and V show the results of using the maximum entropy model with different window sizes of the contexts and POS tags. The model performed well when it used contexts or part-of-speech tags in window size +/-4 words.

In Table VI, the first column indicates which combination of features was used in our model. It is very interesting to see that the system achieves a very low score of the recall when it used all features whereas it achieves a very high score of the precision when it used the combination of contexts, part-of-speech tags, rare word and alphanumeric features and achieves a high score of the recall when it used the combination of contexts, part-of-speech tags, and rare word features. This phenomenon supports that context and part-of-speech information is useful for identifying product features, frequent noun-noun phrases (non-rare words) are likely to be product features and some product features contain letters and numerals such as SD100 and 7-megapixel.

We also compared our approach with baseline. The baseline used only the maximum entropy with the contexts, part-of-speech tags, rare word and alphanumeric features. Our approach used a maximum entropy classifier extracting product features and then applied a natural language processing technique to deal with compound product feature candidates by head word consistency. The performance of the system is shown in Table VII. The baseline achieves 57.48% recall and 78.10% precision. In our approach, the precision is decreased slightly. However, the recall is increased by 17.75% (over 75%). Besides, our approach performs better than using only the maximum entropy model by 7.3% F-score. This result shows that by using an appropriate additional product feature discovery method, our approach can achieve high recall in customer reviews.

We have examined the extraction results manually. It has been found the errors are caused mostly by long and complex sentences. Sometimes people write several sentences without clearly pausing between them. In such cases, the model can not detect sentence boundaries. So the words in the first sentences will be considered as words in the second sentences. This causes the analysis errors in the product feature extraction process. In the future, we can improve our approach by using syntactic dependencies, instead of context windows and adding sentence boundary detection to our model. We also need to increase the size of the training corpus in order to obtain more reasonable feature distributions and parameters. We expect that these improvements will yield an improved product feature extraction task.

VI. CONCLUSION

Review mining and summarization is the task of producing sentiment summary, which consists of sentences from reviews that capture the author's opinion. Product feature extraction is an important task of review mining and summarization. This task is to find out product features that customers refer to in their topic reviews. The goal of our work is to use a machine learning technique to perform automatic product feature extraction. Maximum entropy, a new approach to the task, is applied in order to estimate a function that performs classification of product features and non-product features. The performance of the system with additional product feature discovery can be measured by 71.88% precision, 75.23% recall, and 73.52% F-score. This result shows that the maximum entropy model is effective and suitable to be used for automatic product feature extraction.

REFERENCES

[1] E. Riloff, W. Janyce, and W. Theresa, "Learning subjective nouns using extraction pattern bootstrapping," In Proceedings of ACL SIGNLL Conference, pp. 25-32, 2003.

[2] V. Hatzivassiloglou, and W. Janyce, "Effects of adjective orientation and gradability on sentence subjectivity," In Proceedings of COLING Conference, 2000.

[3] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," In Proceedings of EMNLP Conference, 2002.

[4] P.D. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews," In Proceedings of ACL Conference, pp. 417-424, 2002.

[5] S.M. Kim, and E. Hovy, "Determining the sentiment of opinion," In Proceedings of COLING Conference, pp. 1367–1373, 2004.

[6] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," In Proceedings of HLT/EMNLP Conference, 2005.

[7] M. Hu, and B. Liu, "Mining and summarizing customer reviews," In the Proceedings of the ACM SIGKDD Conference, pp. 168–177, 2004.

[8] A. M. Popescu, and O. Etzioni, "Extracting product features and opinions from reviews," In the Proceedings of the EMNLP Conference, pp. 339–346, 2005.

[9] B. Shi, and K. Chang, "Mining chinese reviews," In the Proceedings of the ICDMW Workshops, 2006.

[10] L. Zhuang, F. Jing, and X. Y. Zhu, "Movie review mining and summarization," In the Proceedings of the CIKM Conference, 2006.

[11] M. Hu, and B. Liu, "Mining opinion features in customer reviews," In the Proceedings of the AAAI Conference, 2004.

[12] B. Liu, M. Hu, and J. Cheng, "Opinion observer: Analyzing and comparing opinions on the Web," In the Proceedings of the WWW Conference, 2005.

[13] G. Carenini, R. T. Ng, and E. Zwart, "Extracting knowledge from evaluative text," In the Proceedings of the K-CAP Conference, pp. 11–18, 2005.

[14] J. Yi, and W. Niblack, "Sentiment mining in WebFountain," In the Proceedings of the ICDE Conference, 2005.

[15] E.T. Jaynes, "Information theory and statistical mechanics," Physics Reviews, vol 106, pp.620-630, 1957.

[16] E.T. Jaynes, Probability theory: The logic of science. Manuscript for book, 1998. http://bayes.wustl.edu/etj/prob.html.

[17] C. Manning, and H. Schutze, Foundations of statistical natural language processing. MIT Press, MA: Cambridge, 1999.

[18] A.L. Berger, D.P. Stephen, and D.P. Vincent, "A maximum entropy approach to natural language processing," Computational Linguistics, vol. 22, no. 1, pp. 39–71, 1996.

[19] J.C. Reynar, and A. Ratnaparkhi, "A maximum entropy approach to identifying sentence boundaries," In the Proceedings of the ANLP Conference, pp. 16-19, 1997.

[20] A. Ratnaparkhi, Maximum entropy models for natural language ambiguity resolution, PhD thesis, University of Pennsylvania, 1998.

[21] H.L. Chieu, and H.T. Ng, "A maximum entropy approach to information extraction from semi-structured and free text," In the Proceedings of the AAAI Conference, pp. 786-791, 2002.

[22] A. Ratnaparkhi, "A maximum entropy model for part-of-speech tagging," In the Proceedings of the EMNLP Conference, pp. 133–141, 1996.

[23] J. Darroch, and D. Ratcliff, "Generalized interative scaling for log-linear model," The Annals of Mathematical Statistics , vol. 43, no. 5, pp. 1470–1480, 1972.

[24] Stanford Lexicalized Parser - A probabilistic lexicalized NL CFG parser, 2006. http://nlp.stanford.edu/downloads/lex-parser.shtml.

[25] J. Baldridge, T. Morton, and G. Bierner, Java-based opennlp maximum entropy package: Maxent. http://maxent.sourceforge.net.