Movie Rating and Review Summarization in Mobile Environment

Chien-Liang Liu, Wen-Hoar Hsaio, Chia-Hoang Lee, Gen-Chi Lu, and Emery Jou

***Abstract*—In this paper, we design and develop a movie-rating and review-summarization system in a mobile environment. The movie-rating information is based on the sentiment-classification result. The condensed descriptions of movie reviews are generated from the feature-based summarization. We propose a novel approach based on latent semantic analysis (LSA) to identify product features. Furthermore, we find a way to reduce the size of summary based on the product features obtained from LSA. We consider both sentiment-classification accuracy and system response time to design the system. The rating and review-summarization system can be extended to other product-review domains easily.**

***Index Terms*—Feature extraction, natural language processing (NLP), text analysis, text mining.**

I. Introduction

PEOPLE's opinion has become one of the extremely important sources for various services in ever-growing popular social networks. In particular, online opinions have turned into a kind of virtual currency for businesses looking to market their products, identify new opportunities, and manage their reputations. Meanwhile, cellular phones have definitely become the most-vital part of our lives. There is no doubt that the mobile platform is currently one of the most popular platforms in the world. However, digital content displayed in cellular phones is limited in size, since cellular phones are physically small. Hence, a mechanism that can provide users with condensed descriptions of documents will facilitate the delivery of digital content in cellular phones. This paper explores and designs a mobile system for movie rating and review summarization in which semantic orientation of comments, the limitation of small display capability of cellular devices, and system response time are considered.

Practically, when we are not familiar with a specific product, we ask our trusted sources to recommend one. Today, the popularity of the Internet drives people to search for other people's opinions from the Internet before purchasing a product or seeing a movie. Many websites provide user rating and commenting services, and these reviews could reflect users' opinions about a product. For example, the customer-review section in Amazon.com lists the number of reviews, the percentage for different ratings, and comments from reviewers. When people want to purchase books, CDs, or DVDs, these comments and ratings usually influence their purchasing behaviors. In addition to these websites, a search engine is another important source for people to search for other people's opinions. When a user enters a query into a search engine, the search engine examines its index and provides a listing of best-matching web pages according to its criteria, usually with a short summary containing the document's title and, sometimes, parts of the text.

Current search engines can efficiently help users obtain a result set, which is relevant to user's query. However, the semantic orientation of the content, which is very important information in the reviews or opinions, is not provided in the current search engine. For example, Google will return around 7 380 000 hits for the query "Angels and Demons review." If search engines can provide statistical summaries from the semantic orientations, it will be more useful to the user who polls the opinions from the Internet. A scenario for the aforementioned movie query may yield such report as "There are 10 000 hits, of which 80% are thumbs up and 20% are thumbs down." This type of service requires the capability of discovering the positive reviews and negative reviews.

In recent years, the problem of "opinion mining" has seen increasing attention [1]–[3]. With the proliferation of reviews, ratings, recommendations, and other forms of online expression, online opinion could provide important information for businesses to market their products, identify new opportunities, and manage their reputations. For example, most recommendation systems attempt to alleviate information overload by identifying which items a user will find worthwhile, and collaborative filtering used in this process relies on the opinions of similar customers to recommend items [4]. Essentially, the task of determining whether a movie review is positive or negative is similar to the traditional binary-classification problem. Given a review, the classifier tries to classify the review into positive category or negative category. However, opinions in natural language are usually expressed in subtle and complex ways. Thus, the challenges may not be addressed by simple text-categorization approaches such as n-gram or keyword-identification approaches [5].

In this paper, we collected movie reviews from Internet Blogs that do not consist of any rating information. Sentiment analysis

Manuscript received July 13, 2010; revised November 4, 2010 and January 18, 2011; accepted February 28, 2011. Date of publication April 29, 2011; date of current version April 11, 2012. This work was supported in part by the National Science Council (NSC) under Grant NSC-99-2221-E-009-150 and Grant NSC-099-2811-E-009-041. This paper was recommended by Associate Editor G. I. Papadimitriou.

C.-L. Liu, W.-H. Hsaio, and C.-H. Lee are with the Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: clliu@mail.nctu.edu.tw; mr.papa@msa.hinet.net; chl@cs.nctu.edu.tw; badlaugh.cs96g@g2.nctu.edu.tw).

G.-C. Lu is with the Global Legal Division iTEC, Hon Hai Precision Industry Company Ltd., Taipei 236, Taiwan (e-mail: badlaugh.cs96g@g2.nctu.edu.tw).

E. Jou is with the Institute for Information Industry, Taipei 106, Taiwan (e-mail: emeryjou@iii.org.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCC.2011.2136334

1094-6977/$26.00 © 2011 IEEE

is performed to determine the semantic orientation of the reviews and movie-rating score is based on the sentiment-analysis result. In addition to the accuracy of the classification, system response time is also taken into account in our system design. Although this paper focuses on movie review, the whole design is not only for movie-review domain. The same design can be applied to other domains such as restaurant, hotel, etc. Meanwhile, increasingly more cellular phones have begun using global positioning system (GPS) functionality, which can utilize user's current location to provide enhanced services and make cellular phones become context aware. Moreover, the opinion-mining result can be used by recommendation systems to identify which items a user will find worthwhile. For example, when people want to have dinner with their friends, restaurant recommendation system can provide a restaurant list based on their current GPS location, opinion-mining result, and their preferences.

In cellular-phone environment, it is inappropriate to display detailed review due to the size of the screen. Hence, we employ summarization technique to reduce the size of information. The system will summarize the reviews (including positive reviews and negative reviews) and provide the user an overview about the reviews. Meanwhile, movie-review summarization is similar to customer review that focuses on product feature [6]. In this paper, we employ feature-based summarization for movie review. Product feature and opinion-word identification are essential to feature-based summarization. We propose an latent-semantic-analysis (LSA) based product-feature-identification approach to identify product features. Moreover, we extend the result to propose an LSA-based filtering mechanism, which can further reduce the size of the summarization according to the features. The main contributions of this paper are the following.

1) Design and develop a movie-rating and review-summarization system in a mobile environment. We considered system response time issue to design the mobile application, and the same system design can be extended to other domains with a little modification.
2) Propose a novel approach based on LSA to identify product features. Product features and opinion words are used to select appropriate sentences to become a review summarization.
3) Propose an LSA-based filtering mechanism to allow the users to choose the features in which they are interested, and this mechanism could reduce the size of summary efficiently.

The rest of this paper is organized as follows. In Section II, related surveys are presented. In Section III, the LSA-based product feature identification approach is introduced. In Section IV, system design is presented. In Section V, several experiments are introduced. In Section VI, the conclusion is presented.

II. Related Surveys

A. Sentiment Analysis

Since a document is composed of sentences and a sentence is composed of terms, it is reasonable to determine the semantic orientation of the text from terms. As a result, the sentiment-analysis research started from the determination of the semantic orientation of the terms. Hatzivassiloglou and McKeown [7] employed textual conjunctions such as “fair and legitimate” or “simplistic but well-received” to separate similarly connoted and oppositely connoted words. Esuli and Sebastiani [3] proposed to determine the orientation of subjective terms based on the quantitative analysis of the glosses of such terms, i.e., the textual definitions that are given in online dictionaries. The process is based on the assumption that terms with similar orientation tend to have “similar” glosses (i.e., textual definitions). Thus, synonyms and antonyms could be used to define a relation of orientation. Esuli and Sebastiani [8] described SENTIWORDNET, which is a lexical resource in which each WordNet synset is associated with three numerical scores, i.e., Obj(s), Pos(s), and Neg(s), thus describing how objective, positive, and negative the terms contained in the synset.

Traditionally, sentiment classification can be regarded as a binary-classification task [1], [2], [9]. Turney [2] proposed to determine the orientation of terms by bootstrapping from a pair of two minimal sets of “seed” terms by counting the number of hits returned from search engine with a $NEAR$ operator. The $NEAR$ operator requires these two phrases or terms to be within a specified word count of one another to be counted as a successful result. AltaVista search engine[1] allows the user to specify a word distance of his/her choice, but the maximum distance is ten words. The relationship between a given phrase and a set of seeds was used to place it into a positive or negative subjectivity class. Pang *et al.* [1] found out that standard machine learning outperforms human-proposed baselines. They employed naive Bayes, maximum-entropy classification, and support vector machines (SVMs) [10] to perform sentiment-classification task on movie-review data. According to their experiment, SVMs tended to do the best, and unigram with presence information turns out to be the most effective feature.

In recent years, some researchers have extended sentiment analysis to the ranking problem, where the goal is to assess review polarity on a multipoint scale [11]–[13]. Snyder and Barzilay [13] addressed the problem of analyzing multiple related opinions in a text and presented an algorithm that jointly learns ranking models for individual aspects by modeling the dependencies between assigned ranks. Goldberg and Zhu [12] proposed a graph-based semisupervised learning algorithm to address the sentiment-analysis task of rating inference and their experiments showed that considering unlabeled reviews in the learning process can improve rating inference performance.

B. Feature-Based Summarization

In product-review summarization, people are interested in the reasons why this product is worth buying rather than the principal meaning of the comment. Thus, feature-based summarization [6] is used in movie-review summarization. The feature-based summarization will focus on the product features on which the customers have expressed their opinions. In addition to product features, the summarization should include

[1] AltaVista: http://http://www.altavista.com/

opinion information about the product; therefore, product features and opinion words are both important in feature-based summarization. As a result, product features and opinion-word identification are essential in feature-based summarization.

Practically, it is not easy to list all the product features and opinion words manually. Some researchers try to use a statistical approach to identify frequent feature words, because the product features may occur frequently in product reviews. However, the drawback of this approach is that it may miss infrequent features. Hu and Liu [6] studied the problem of generating feature-based summaries of customer reviews of products sold online and proposed a method of word attributes, including occurrence frequency, part of speech (POS), and synset in WordNet. Meanwhile, Zhuang *et al.* [14] proposed to make use of grammatical rules and keyword lists to seek for feature-opinion pairs and generate feature-based summarization. Lu *et al.* [15] utilized POS tagging and chunking function of the OpenNLP[2] toolkit to identify phrases in the form of a pair of head term and modifiers. Their research focused on short comments; therefore, POS-tagging information can be employed to obtain the product features and opinion words. For example, the comment "Fast ship and delivery" contains only one sentence; therefore, it is easier to obtain the head terms (i.e., noun or noun phrase) and modifiers (i.e., adjective) using POS-tagging information. Practically, this approach cannot be applied to other product-review applications. First, most reviews contain many sentences rather than short comments. Second, most sentences in a review often contain many terms that are irrelevant to the product features or opinion words. Thus, we cannot identify the product features and opinion words in movie reviews using the same approach.

III. Latent-Semantic-Analysis-Based Product-Feature Identification

In this paper, we propose a novel approach based on LSA to identify related product-feature terms. Essentially, LSA is a theory and method to analyze relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. LSA can be applied to any type of count data over a discrete dyadic domain, which is so-called two-mode data [16]. Supposing that a collection of documents $D = \{d_1, \ldots, d_n\}$ with terms from $W = \{w_1, \ldots, w_m\}$ are given, then the system can construct a cooccurrence matrix M, where its dimension is $n \times m$ and each entry M_{ij} denotes the number of times the term w_j occurred in document d_i. Each document d_i is represented using a row vector, while each term w_j is represented using a column vector. As shown in (1), LSA applies singular-value decomposition (SVD) to the term-document matrix M, and a low-rank approximation of the matrix M could be used to determine patterns in the relationships between the terms and concepts contained in the text

$$M = U\Sigma V^T \tag{1}$$

Algorithm 1: LSA-based Product Feature Identification Algorithm

```
   Input: A $n \times m$ term-document matrix $M$, product
          feature seed set $S$, reduced dimension $k$, the
          number of extracted features for each seed $n$
   Output: An association array $F$, where each key
           represents a product feature seed $f$ and its
           corresponding value is $f$'s related product
           features
 1 begin
 2     initialize associated array $F$
 3     $U, \tilde{\Sigma}, V^t \longleftarrow$ svd($M$, $k$)
 4     $\tilde{M} \longleftarrow U \times \tilde{\Sigma} \times V^t$
 5     for $f \in S$ do
 6         $w_f \longleftarrow$ getTermVectorFromTermDocMatrix($f$,
           $\tilde{M}$)
 7         initialize similarity list $sim$
 8         $i \longleftarrow 1$
 9         foreach column vector $w$ of $\tilde{M}$ do
10             $sim[i] \longleftarrow w_f \cdot w$
11             $i \longleftarrow i + 1$
12         end
13         sort($sim$)
14         $relatedFeatureList \longleftarrow$
           getTopRelatedFeatures($sim, n, \tilde{M}$)
15         $F[f] \longleftarrow relatedFeatureList$
16     end
17     return $F$
18 end
```

where U and V are matrices with orthonormal columns (i.e., $U^T U = V^T V = I$), and Σ is a diagonal matrix whose diagonal elements are the singular values of M.

The original term-document matrix could be approximated by reducing the dimensions of the term–document space, and this will allow the underlying latent relationships between terms and documents to be exploited during searching. Equation (2) shows that the reduced matrix $\tilde{M}$ is obtained by reducing the dimensionality, where the system truncates the singular-value matrix Σ to size k. It is this dimensionality-reduction step, i.e., the combining of surface information into a deeper abstraction, which captures the mutual implications of words and passages. Therefore, even though the original vector space is sparse, the corresponding low-dimensional space is typically not sparse. Practically, the number of dimensions retained in LSA is an empirical issue [17]. We conducted the experiments under different dimensions in the experiment section

$$\tilde{M} = U\tilde{\Sigma}V^T \approx U\Sigma V^T = M. \tag{2}$$

Algorithm 1 shows the algorithm, where the inputs include a term-document matrix, several product-feature seeds, the reduced dimensionality in SVD operation, and the number of extracted features for each seed. In Algorithm 1, lines 3 and 4 are employed to perform linear algebra SVD operation on the term-document matrix, and lines 5–16 are used to compute the similarities between the seed product-feature vector and,

[2] http://opennlp.sourceforge.net/

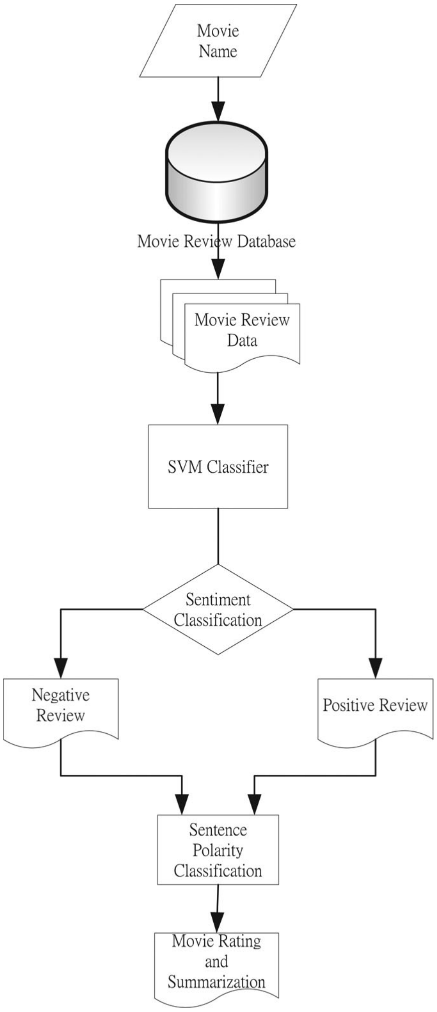


Fig. 1. Movie review and summarization flow.

pairwise, the other term vectors. The top ones will be collected as related product-feature terms for a specific product feature. The procedure **getTermVectorFromTermDocMatrix** is used to obtain the term-vector representation of a product feature. The seed is supposed to be one of the terms in the term-document matrix, and it is easy to obtain its corresponding document-vector representation. Meanwhile, sim in line 7 is used to store the similarities between the seed and the other terms. After sorting in descendant order, it is easy to obtain the top ones and their corresponding feature names in procedure **getTopRelatedFeatures**.

When the above steps are completed, each product-feature seed can have its own semantically related term set. The advantage of this approach is that it could be applied to all the languages, it does not need any external dictionary, since LSA is language-independent, and it is based on linear algebra SVD operation.

IV. System Design

Fig. 1 shows the system flow. The input is a movie name and the system will use the movie name to retrieve reviews about this movie from movie-review database. These movie reviews become the inputs of the SVM sentiment classifier, which will classify the reviews into positive or negative classes. Rating information can be obtained based on the proportion of positive and negative movie reviews. In addition to the sentiment classification of movie review, we further determine the polarity of a sentence using opinion words. Then, the system can provide both positive and negative summarization, regardless of the polarity of a review. The whole process includes sentiment classification and feature-based summarization. These two processes will be described in the following sections.

A. Dataset

In this paper, we collected the Chinese movie reviews from Internet Blogs. Since the original data are an hypertext markup language (HTML) document, HTML-tag-removal process is required to extract the text information. Training data are necessary for SVM to train a classification model, and manual classification is performed to classify the training reviews into positive or negative reviews. We randomly selected 500 positive reviews and 500 negative reviews as the data for classification-model building. In addition to the model-building data, we further collected around 8000 movie reviews from the Internet, and these reviews will be used as movie-review database.

B. Sentiment Classification

As mentioned above, sentiment classification is similar to traditional binary-classification problem. Currently, many classification algorithms such as SVM [1], [10], [18], [19], decision trees [20], and neural networks [21] have been proposed and shown their capabilities in different domains. SVM is one of the state-of-the-art algorithms. SVM has been shown to be highly effective in traditional text categorization. SVM measures the complexity of hypotheses based on the margin with which they separate the data instead of the number of features. One remarkable property of SVM is that their ability to learn can be independent of the dimensionality of the feature space.

In natural-language processing (NLP) and information retrieval (IR), bag-of-words model tries to use an unordered collection of words to represent a text, disregarding grammar and even word order. In other words, each word in the text contributes to a feature of the document. In this paper, we employ similar approach to construct a feature vector of the document. Stop words are removed first and then each distinct word W_i in the document is used to represent a feature. As a result, a document could be represented by a feature vector, and many machine-learning algorithms could be applied to perform classification tasks. We employed SVM to perform the classification and libsvm [22] package is used in the system. The kernel function used in the system is the radial basis function (RBF) and K-fold cross validation (i.e., $K = 5$) is conducted in the experiment.

The classification result will be the basis of the rating. With the proportion of positive and negative reviews, the system could provide the rating information to end users. For example, if there are 100 movie reviews for a specific movie and 80 reviews are positive, the rating of this movie will be four stars.

C. Review Summarization

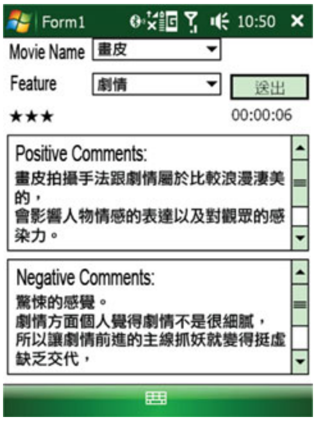
1) Product-Feature Identification: As mentioned above, we propose an LSA-based product-feature-identification algorithm and system can obtain a semantically related feature set for each seed. We compared three product-feature-identification approaches, i.e., the LSA-based approach, frequency-based approach, and PLSA-based approaches, in the experiment section.

2) Opinion-Word Identification: In addition to feature identification, opinion words about the product features are important as well. Hu and Liu [6] extracted the opinion words by retrieving the nearby adjective of product features. In addition to language sentence-structure characteristic, Zhuang *et al.* [14] used the dependency grammar graph to find out some relations between feature words and the corresponding opinion words in training data. They both rely on language sentence structure to extract opinion words; therefore, these approaches will be applicable to those language sentences having such a characteristic.

Many languages do not possess the aforementioned sentence structure. Hence, we propose to use a statistical approach to discover opinion words. First, we take into account POS-tagging information of the opinion words. According to our analysis, adjectives are usually used to describe sentiment in Chinese; therefore, these terms become the candidate opinion words. Second, term frequency is taken into account; therefore, frequency of the opinion words should exceed a threshold value. Let AVG be the average of sum of square of frequency of all items as shown in (3) below. A term$_i$ will be selected only if its square of frequency is equal or larger than AVG. We manually selected positive and negative sentences from 500 positive reviews and 500 negative reviews, respectively. Positive opinion words and negative opinion words could be further obtained based on term frequency and POS tagging.

$$\begin{aligned} S_f &= \sum_{i=1}^{n} \{\text{Frequency}(\text{term}_i)\}^2 \\ \text{AVG} &= S_f/n. \end{aligned} \tag{3}$$

3) Feature-Based Summarization: As described above, feature-based summarization is more appropriate in product-review summarization. In general, feature-based summarization is based on product features and opinion words. It is not easy to use compression ratio directly, since the sentence-selection criterion is based on the presence of product features. Hence, we propose an LSA-based filtering approach to further select the content of the summary based on user's favor. In product-feature discovery, we employ LSA to find out related feature terms of a specific product feature, and these related terms could be regarded as being semantically related to this product feature. For each given product feature f, LSA could discover related terms F that are semantically related to f. In general, F could be regarded as f's related terms, and the system can employ F to select summary sentences. In application design, the system provides all the summary sentences in the beginning. The product-feature seeds mentioned in LSA-based feature-identification process will become candidate interested features. The system allows the user to determine the feature f in which he/she is interested. When the user determines f, the system will generate a summary, which is related to product features F.



Movie Name 畫皮

Feature 劇情

登出

★★★ 00:00:06

Positive Comments:
畫皮拍攝手法跟劇情屬於比較浪漫淒美的，
會影響人物情感的表達以及對觀眾的感染力。

Negative Comments:
驚悚的感覺。
劇情方面個人覺得劇情不是很細膩，
所以讓劇情前進的主線抓妖就變得挺虛缺乏交代，

Fig. 2. Rating and summarization screenshot.

Practically, a positive movie review may include negative comments about specific aspects and *vice versa*. In this paper, we propose to analyze the polarity of a movie review using SVM and analyze the polarity of a sentence using opinion words. In feature-based summarization, the system can utilize the polarity of opinion words to determine the polarity of sentences. Hence, the system can provide both positive- and negative-review summarization, regardless of the polarity of a review.

With the proportion of positive and negative reviews, the system could provide the rating information to end users. The rating information combined with review summary could give end users the rating and summarization information about the movie. The "Feature" section in Fig. 2 is a pull-down menu, which allows the users to choose the features in which they are interested. Meanwhile, positive summarization and negative summarization can be presented to users, regardless of a movie's rating.

V. Experiment

Several experiments are performed to evaluate our system. In sentiment-classification experiment, SVM is employed to perform the sentiment-classification task. Several feature combinations are used to evaluate the system performance. Since the application runs on mobile platform, therefore, classification accuracy is not the only factor in system design. The system will be infeasible if it takes a long time to response. Therefore, system-response-time-evaluation experiment is conducted as well. In product-feature identification, we propose an LSA-based approach to identify the product features and compare LSA-based approach with frequency-based and PLSA-based approaches using the movie-review-glossary dataset.

A. Sentiment Classification

Opinions in natural language are usually expressed in subtle and complex ways. For example, the polarity of a sentence may be changed when a negative term is used in the sentence. We considered possible feature combination in the experiments to obtain the best feature selection. Based on the bag-of-words model, we used unigram, bigram, negation, location, frequency, and presence features (i.e., only consider whether the feature is present or not) to perform the classification task with different feature combinations.

Unlike English, the Chinese language could not make use of spaces as a boundary to separate words in a sentence. Chinese-word-segmentation process is required. In addition to Chinese-word segmentation, Chinese stop words are removed as well, since the stop words cannot provide sufficient information. In feature selection, our experiments also showed that unigram with presence features outperforms bigram with other features, and the result is the same as described in [1]. In addition to unigram with presence features, we design three basic experiments to compare the differences of feature combinations, and they are described as follows.

1) Group 1:
   a) removal of the terms appearing in both positive and negative reviews;
   b) frequency-feature criterion, where the term's square of frequency should be at least AVG, as shown in (3);
2) Group 2: frequency-feature criterion, where the term's square of frequency should be at least AVG, as shown in (3);
3) Group 3: frequency-feature criterion, where the term should occur at least three times.

The Group 1 experiment includes two additional features to evaluate its performance. The first feature is about the removal of the terms appearing in both positive and negative reviews. In general, the terms that appear in both positive and negative reviews could not provide enough semantic orientation to differentiate positive and negative reviews. The second feature is about the comparison of the effect of frequency.

The Group 1 and Group 2 experiments are used to compare the effect of term selection. While Group 1 removed the terms appearing in both positive and negative reviews, the Group 2 experiment used all the terms. The Group 2 and Group 3 experiments are used to compare the effect of term frequency. While Group 2 used the frequency criterion based on (3), Group 3 selected the terms that occur at least three times.

These three experiments are performed to evaluate their performances on movie-review data, and they will become the bases of other experiments. Negation and position are additional features that are included into these three bases to perform feature combination. In negation feature, a negation term may change the polarity of a sentence completely, which may blur the decision. For example, a sentence "This movie is interesting" indicates a positive opinion about this movie, while the sentence "This movie is not interesting" changes the polarity of the sentence. As for position feature, people may have the conclusion in the end, therefore, position feature is employed, as well to evaluate its effect.

TABLE I
EXPERIMENT RESULT

Feature	Accuracy
Unigram with presence feature	85.40%
Group-1	71.00%
Group-1 + Negation	70.79%
Group-1 + Position	69.51%
Group-2	78.46%
Group-2 + Negation	79.32%
Group-2 + Position	71.64%
Group-3	76.55%
Group-3 + Negation	75.48%
Group-3 + Position	70.15%

TABLE II
SVM MODEL LOADING AND PREDICTION EVALUATION RESULT

SVM Model Loading and Prediction Evaluation (sec)			
Feature type	Number of features	Model Loading	Prediction
Frequency-based	1,902	5.328	0.015 - 0.0625
Unigram with presence	40,462	119.5	0.5 - 0.625

Table I shows the experimental result. Unigram with presence feature (i.e., only considers the presence and absence of a term) outperforms the other feature combinations, and this result conforms to Pang's [1] result. It seems like that negation, location, and bigram features do not contribute to sentiment classification. If we compare the performance of three basic experiments, Group 2 outperforms Group 1 and Group 3. In other words, the removal of the terms appearing in both positive and negative reviews will decrease the classification-accuracy rate. Meanwhile, the frequency criterion based on (3) is a little better than the frequency criterion, which is at least three times. Furthermore, the feature-combination experiments show that Group 2 with negation feature outperforms Group 2, and this result is different from Pang's [1] research result.

However, sentiment-classification accuracy is not the only issue on mobile platform, and response time should be considered as well. Table II shows that the system using unigram with presence feature will have 40 462 features, and it takes about 120 s to load the classification model. Obviously, it is infeasible on mobile platform if a system's response takes 120 s. Hence, the number of features is crucial to the system's response time. We employ frequency as filtering criterion to reduce the number of features. The number of features could be reduced to 1902 if we use the frequency criterion based on (3). Table II shows that it takes about 6 s to load classification model, and it is feasible on mobile platform. Therefore, this frequency criterion is employed to perform sentiment classification.

We also performed sentiment classification on another movie-review dataset, which is available at http://www.cs.cornell.edu/People/pabo/movie-review-data/. The dataset includes 1000 positive and 1000 negative movie reviews. Similarly, SVM is used to perform the classification task. The kernel function used in the system is RBF and K-fold cross validation (i.e., $K = 5$) is used in the experiment. Different feature-selection criteria are used in the experiment to compare their number of features and accuracies. Table III shows the experimental result, which includes three feature-selection approaches. The preprocess task includes the punctuation-elimination process, the lowercase-conversion process, and the negative-term-conversion process, which converts "n't" to "not." The first one used all the unigrams as features, while the second one

TABLE III
SENTIMENT-CLASSIFICATION RESULTS USING PUBLIC MOVIE-REVIEW DATASET

Feature Selection Criterion	Number of Features	Accuracy
Unigrams	36,084	86.5%
Unigrams with occurrences more than 3	15,026	86.25%
Unigrams using the frequency criterion based on Equation (3)	861	81.2%

employed frequency as the filtering criterion, with only the unigrams with occurrences more than three would be taken into account. The third one employed the frequency criterion listed in (3). The term-document matrices of all the experiments employed unigram with presence feature as entry value. The first two approaches do not remove stop words, but the third one removes stop words first. The main reason is that stop words are the terms with high frequencies, therefore, almost only stop words will be left using the criterion listed in (3) if the stop words are not removed in advance of the process.

The experimental results are similar to the previous experiment. The first one outperforms the other ones, but the number of features is enormous. The second one can reduce more than half of the features and the accuracy is almost the same. However, the number of features is still enormous. The number of features in the third experiment is 861 and its accuracy is about 81.2%. Although the accuracy of the third one is not as good as the other ones, it can dramatically reduce the number of features. Meanwhile, its accuracy is still acceptable practically.

B. Product-Feature Identification

In product-feature identification, we compared our LSA-based approach with two other approaches, which are frequency-based and PLSA-based. We performed experiments using the movie-review documents mentioned above, which is available at http://www.cs.cornell.edu/People/pabo/movie-review-data/. The dataset includes 1000 positive and 1000 negative movie reviews. Since nouns are the candidates of product features, only nouns will be used in this experiment and the total number of nouns is 29 632. In addition to movie-review dataset, we employed the movie-review glossary, which is available at http://www.movieprofiler.com/movieglossary, as the basis of the comparison. The movie-review glossary is created for movie reviewers, critics, and film students alike, as well as the general public interested in movie reviewing and film making-related terminology. The number of terminologies is 1069. Since many terminologies are only used in movie industry, additional filtering is applied to the dataset. Only the terms appearing in the movie-review data will be kept. The number of terminologies left is 383. A copy of the terminologies obtained from movieprofiler.com and the terminologies used in this paper are available at http://islab.cis.nctu.edu.tw/download/. Precision, recall, and F-value are employed to evaluate system performance.

In frequency-based approach, all the nouns are ranked according to their frequencies, and then, the top ones are selected as product features. Table IV shows the top ten terms using frequency-based approach. Frequency-based approach can identify the terms that are often used in movie reviews. Hence, the terms like story, character, and plot can be identified. In the LSA-based approach, Algorithm 1 is used to identify product features and the seeds include scene, plot, director, actor, and story. The truncated dimension of LSA is 500 in this paper. Table V shows the top ten features for each seed. In addition to product-feature identification, the top ten features for each seed can be regarded as being semantically related to the seed.

TABLE IV
TOP TEN TERMS USING FREQUENCY-BASED APPROACH

Ranking	Terms
1	film
2	movie
3	time
4	story
5	films
6	characters
7	character
8	life
9	plot
10	people

TABLE V
FIVE ASPECTS GENERATED USING LSA

Scene	Plot	Director	Actor	Story
film	film	film	film	film
movie	movie	movie	movie	story
scene	plot	director	actor	movie
time	time	films	character	characters
films	times	time	time	films
character	characters	story	films	time
characters	character	characters	story	character
story	story	character	characters	life
scenes	action	scene	scene	people
people	movies	plot	plot	scene

In PLSA-based approach, we applied PLSA [23] to the dataset. Essentially, PLSA is based on a mixture decomposition derived from a latent class model. The standard procedure for maximum-likelihood estimation in latent-variable models is the expectation–maximization (EM) algorithm [24], which includes the E-step and the M-step. In E-step, the posterior probabilities are computed for the latent variable z based on the current estimates of the parameters. In M-step, the parameters are updated based on the posterior probabilities obtained in the previous E-step. When given each occurrence of a word $w \in W = \{w_1, \dots, w_M\}$ in a document $d \in D = \{d_1, \dots, d_N\}$, the E-step is given by

$$P(z_k|d_i, w_j) = \frac{P(w_j|z_k)P(z_k|d_i)}{\sum_{l=1}^{K} P(w_j|z_l)P(z_l|d_i)}. \tag{4}$$

TABLE VI
FIVE ASPECTS GENERATED USING PLSA

Aspect 1	Aspect 2	Aspect 3	Aspect 4	Aspect 5
film	film	film	movie	film
movie	movie	movie	story	movie
action	comedy	time	director	action
jackie	time	story	review	story
time	plot	characters	job	films
plot	sex	films	sex	van
films	scene	character	gauge	time
scenes	story	life	granger	characters
director	words	people	grangers	plot
character	films	movies	comedy	scenes

The estimate $P(d_i) \propto n(d_i)$ can be carried out independently. By standard calculation, one arrives at the following M-step reestimation equations. The number of aspects is five in PLSA experiment, and Table VI shows the top ten terms for each aspect

$$P(w_j|z_k) = \frac{\sum_{i=1}^{N} n(d_i, w_j) P(z_k|d_i, w_j)}{\sum_{m=1}^{M} \sum_{i=1}^{N} n(d_i, w_m) P(z_k|d_i, w_m)} \quad (5)$$

$$P(z_k|d_i) = \frac{\sum_{j=1}^{M} n(d_i, w_j) P(z_k|d_i, w_j)}{n(d_i)}. \quad (6)$$

Furthermore, frequency-based, LSA-based, and PLSA-based approaches are applied to the movie-review dataset, and the terms extracted from these approaches are compared with the terms in filtered movie glossary dataset. Fig. 1 shows the result, where precision, recall, and F-value curves are presented. In LSA and PLSA approaches, many terms may appear in different aspects; therefore, performance evaluation only takes into account distinct terms. In other words, the term "film" in LSA and PLSA will be calculated once. The experimental results show that LSA outperforms frequency-based and PLSA-based approaches in precision, recall, and F-value evaluations. As a by-product, the system can identify a related term set for each seed. Meanwhile, as shown in Fig. 3, PLSA-based approach does not work well in product-feature identification.

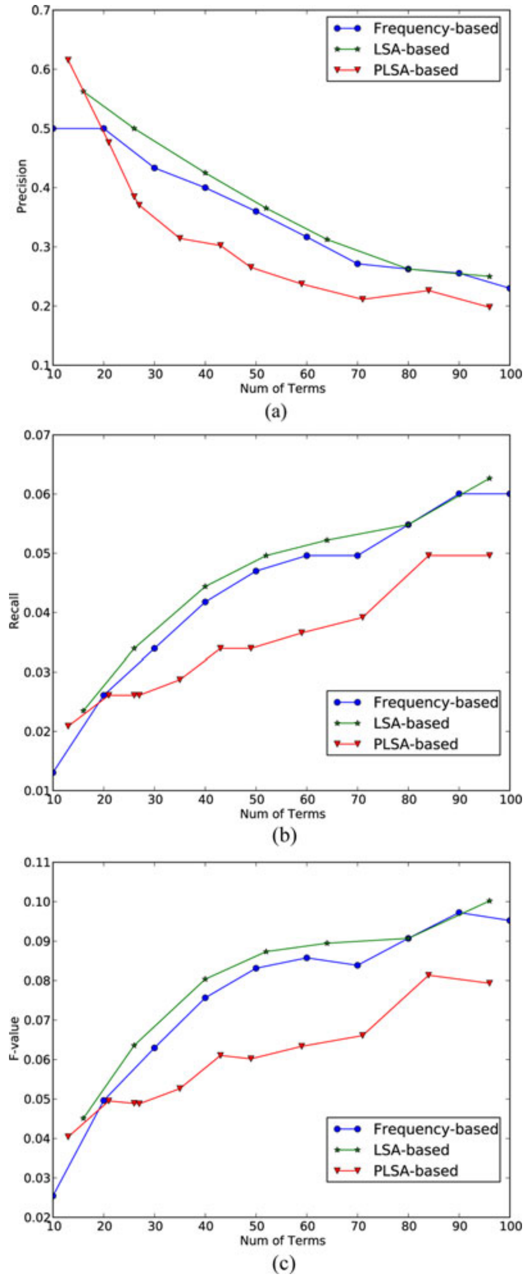


Fig. 3. Precision, recall, and F-value curves for movie-review-glossary dataset. (a) Precision curve. (b) Recall curve. (c) F-value curve.

In addition to the experiments mentioned above, we further conducted experiments on the effect of truncated dimension of LSA in product-feature identification. We conducted the experiments under different dimensions and compared the results with the frequency-based approach. Fig. 4 shows the result, where precision, recall, and F-value curves are presented. As shown in Fig. 4, LSA outperforms frequency-based approach when the number of dimensions is more than 500. For LSA, the differences are minor when the number of dimensions is more than 500. On the other hand, if the number of dimensions is 50, the performance becomes worse than the frequency-based approach when the number of terms is more than 80.

Basically, PLSA can be regarded as a clustering algorithm. As shown in the above experiment, PLSA cannot work well on the movie-review dataset. To further investigate the clustering capability of PLSA, we performed another experiment on a popular dataset, which is 20 newsgroups dataset. The 20 newsgroups collection has become a popular dataset for experiments in text applications of machine-learning techniques, such as text classification and text clustering. The data are organized into 20 different newsgroups, each corresponding to a different topic. Besides PLSA, we also applied LSA and k-means algorithms to the same dataset for comparison. In LSA approach, dimensionality-reduction process is performed first (i.e., the dimensionality of $\tilde{\Sigma}$ is 300), then k-means-clustering algorithm is applied to reduced matrix $\tilde{M}$. We used three newsgroups, which include alt.atheism, comp.graphics, and comp.sys.ibm.pc.hardware, from the dataset to evaluate the clustering performance.

We compared the generated clusters by using the F1 cluster-evaluation measure [25]. The F1 cluster-evaluation measure

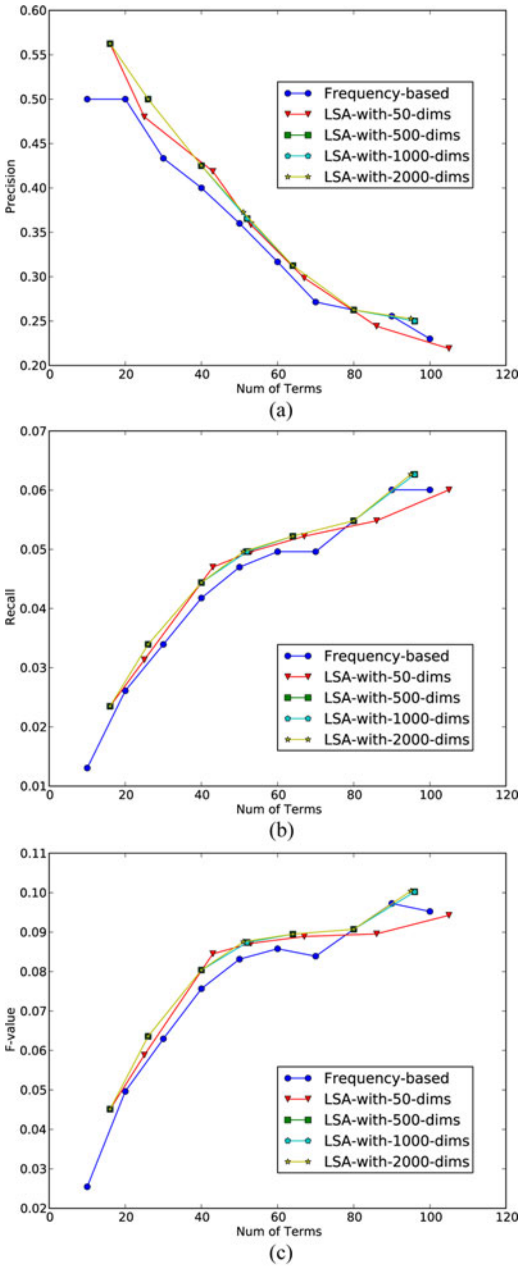


Fig. 4. Precision, recall, and F-value curves for movie-review-glossary dataset using LSA under different truncated dimensions. (a) Precision curve. (b) Recall curve. (c) F-value curve.

considers both precision and recall, where precision and recall here are computed over pairs of documents for which two label assignments either agree or disagree. The F1 cluster-evaluation measure is also used by Ramage *et al.* [26]. The following four evaluation metrics are necessary for the computation.

1) *True positives (TPs):* The clustering algorithm placed the two articles in the pair into the same cluster, and 20 newsgroups have them in the same class.
2) *False positives (FPs):* The clustering algorithm placed the two articles in the pair into the same cluster, but 20 newsgroups have them in differing classes.

TABLE VII
CLUSTERING RESULT USING 20 NEWSGROUPS DATASET

	Precision	Recall	F1 value
k-means	0.4820	0.5645	0.5200
LSA	0.5096	0.5378	0.5233
PLSA	0.8301	0.8363	0.8332

TABLE VIII
THREE ASPECTS GENERATED USING PLSA (20 NEWSGROUPS DATASET)

Aspect 1	Aspect 2	Aspect 3
edu	edu	edu
thank	com	write
file	drive	com
post	us	articl
graphic	card	on
image	thank	god
us	on	post
know	know	don
anyon	system	think
program	post	peopl

3) *True negatives (TNs):* The clustering algorithm placed the two articles in the pair into differing clusters, and 20 newsgroups have them in differing classes.
4) *False negatives (FNs):* The clustering algorithm placed the two articles in the pair into differing clusters, and 20 newsgroups have them in the same class.

Similar to the traditional IR definition, (7) shows the formulas of precision, recall, and F1 evaluation

$$\begin{aligned}\text{Precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}}\\ \text{Recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}}\\ \text{F1} &= \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}.\end{aligned}\tag{7}$$

Table VII shows the experimental results, where PLSA outperforms k-means and LSA. The PLSA works very well in the clustering of newsgroups dataset. Moreover, Table VIII shows the top ten terms of the aspects discovered by PLSA. Obviously, these three newsgroups are highly unrelated, and we can determine their clusters from their top ten terms. The aspect 1 belongs to comp.graphics newsgroup, the aspect 2 belongs to comp.sys.ibm.pc.hardware newsgroup, and the aspect 3 belongs to alt.atheism newsgroup. On the other hand, it is very difficult to distinguish the aspects of movie-review dataset. The plausible reason might be that the articles in movie-review dataset are all about movie reviews, and most reviewers may use similar terms in their articles.

C. Discussion

In sentiment classification, Pang *et al.* [1] showed that unigram with presence features outperformed other feature combinations. Our experiments conform to Pang's research results. However, if all the unigrams are used in the system, the number of features will be enormous. For example, our training dataset includes 1000 movie reviews, and the number of features is around 40 000. The application needs to load SVM model first and then predict the semantic orientation of the review. If 40 000 features are used, it would take around 120 s to load the model. Hence, we employed frequency criterion to reduce the number of features. Currently, our system uses 1902 features, and it takes less than 6 s to load model and predict the review.

In product-feature identification, the experiment shows that LSA-based approach outperforms frequency-based and PLSA-based approaches. As a by-product, our LSA-based system can identify a related term set for each seed. We propose an LSA-based filtering mechanism to employ these semantically related terms to reduce the size of summary. Only the sentences containing these terms will be presented to users. Moreover, the LSA-based product-feature-identification approach could be generalized to other product-review domains, since the linear algebra SVD operation could be applied to any language.

Meanwhile, we conducted an experiment on the truncated dimension of LSA. Several truncated-dimension values were used, and their results were compared with frequency-based approach. The experimental result shows that when the truncated dimension is more than 500, the differences are minor.

Moreover, we used 20 newsgroups dataset to evaluate PLSA's clustering performance. The result shows that PLSA could outperform k-means and LSA. One of the important features of the newsgroup dataset is that the newsgroups in the experiment are highly unrelated. In other words, the boundaries between these aspects are very clear. However, the movie-review dataset does not possess such a characteristic. The articles in the movie review are similar, since they all focus on movie reviews. Hence, it might be the reason why PLSA could not determine the boundaries between the aspects of movie reviews.

Currently, feature-based summarization is sentence-level summarization. Although summary sentences are about product features and opinion words, these sentences are obtained from different paragraphs or movie reviews. It is obvious that a fluency problem exists in the summary. Thus, it will be our future work to achieve greater fluency of the summarization.

VI. Conclusion

In this paper, we design and implement a movie-rating and review-summarization system in mobile environment. Sentiment classification is applied to the movie reviews, and rating information is based on sentiment-classification results. In feature-based summarization, product-feature identification plays an essential role, and we propose a novel approach based on LSA to identify related product features. Moreover, we use a statistical approach to identify opinion words. Product features and opinion words will be used as the basis for feature-based summarization.

In a system-performance-analysis experiment, the number of features plays an important role in SVM-model loading and prediction. We use frequency criterion to reduce the number of features, and the experiment shows that it takes less than 6 s to load the SVM model and classify the reviews. Furthermore, we propose an LSA-based filtering approach to reduce the size of the summary based on the user's preferred aspect. The design proposed in this paper could fully utilize the Internet content to provide a new product-review summarization and rating service. The design can also be extended to other product-review domains easily.

References

[1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proc. ACL-02 Conf. Empirical Methods Natural Lang. Process.*, 2002, pp. 79–86.

[2] P. D. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguist.*, 2002, pp. 417–424.

[3] A. Esuli and F. Sebastiani, "Determining the semantic orientation of terms through gloss classification," in *Proc. 14th ACM Int. Conf. Inf. Knowl. Manage.*, 2005, pp. 617–624.

[4] S. H. Choi, Y.-S. Jeong, and M. K. Jeong, "A hybrid recommendation method with reduced data for large-scale application," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 5, pp. 557–566, Sep. 2010.

[5] T. Mullen and N. Collier, "Sentiment analysis using support vector machines with diverse information sources," in *Proc. EMNLP*, 2004, pp. 412–418.

[6] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2004, pp. 168–177.

[7] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in *Proc. 8th Conf. Eur. Chap. Assoc. Comput. Linguist.*, Morristown, NJ: Assoc. Comput. Linguist., 1997, pp. 174–181.

[8] A. Esuli and F. Sebastiani, "SENTIWORDNET: A publicly available lexical resource for opinion mining," in *Proc. 5th Conf. Lang. Res. Eval.*, 2006, pp. 417–422.

[9] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: opinion extraction and semantic classification of product reviews," in *Proc. 12th Int. Conf. World Wide Web*, New York: ACM, 2003, pp. 519–528.

[10] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[11] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proc. 43rd Annu. Meet. Assoc. Comput. Linguist*, Morristown, NJ: Assoc. Comput. Linguist., 2005, pp. 115–124.

[12] A. B. Goldberg and X. Zhu, "Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization," in *Proc. TextGraphs: First Workshop Graph Based Methods Nat. Lang. Process*, Morristown, NJ: Assoc. Comput. Linguist., 2006, pp. 45–52.

[13] B. Snyder and R. Barzilay, "Multiple aspect ranking using the good grief algorithm," in *Proc. HLT-NAACL*, 2007, pp. 300–307.

[14] L. Zhuang, F. Jing, and X.-Y. Zhu, "Movie review mining and summarization," in *Proc. 15th ACM Int. Conf. Inf. Knowl. Manage.*, 2006, pp. 43–50.

[15] Y. Lu, C. Zhai, and N. Sundaresan, "Rated aspect summarization of short comments," in *Proc. 18th Int. Conf. World Wide Web*, New York: ACM, 2009, pp. 131–140.

[16] T. Hofmann, J. Puzicha, and M. I. Jordan, "Learning from dyadic data," in *Proc. Conf. Adv. Neural Inform. Process. Syst. II*, Cambridge, MA: MIT Press, 1999, pp. 466–472.

[17] T. K. Landauer, P. W. Foltz, and D. Laham, "Introduction to latent semantic analysis," *Discourse Processes*, vol. 25, pp. 259–284, 1998.

[18] T. Joachims, *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Norwell, MA: Kluwer, 2002.

[19] C. Silva, U. Lotrič, B. Ribeiro, and A. Dobnikar, "Distributed text classification with an ensemble kernel-based learning approach," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.*, vol. 40, no. 3, pp. 287–297, May 2010.

[20] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers—A survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 35, no. 4, pp. 476–487, Nov. 2005.

[21] G. P. Zhang, "Neural networks for classification: A survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 30, no. 4, pp. 451–462, Nov. 2000.

[22] (2001). *LIBSVM: A library for support vector machines* [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[23] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Mach. Learn.*, vol. 42, no. 1/2, pp. 177–196, 2001.

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin. (1977). Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc., Series B* [Online]. vol. 39, no. 1, pp. 1–38. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.133.4884.

[25] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. New York: Cambridge Univ. Press, 2008.

[26] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia-Molina, "Clustering the tagged web," in *Proc. 2nd ACM Int. Conf. Web Search Data Mining*, New York: ACM, 2009, pp. 54–63.

Chien-Liang Liu received the M.S. and Ph.D. degrees in computer science from National Chiao Tung University, Hsinchu, Taiwan, in 2000 and 2005, respectively.

He is currently a Postdoctoral Researcher with the Department of Computer Science, National Chiao Tung University. His current research interests include machine learning, natural-language processing, and data mining.

Wen-Hoar Hsaio received the B.S. degree from the Department of Computer Science and Information Engineering, Chung Cheng Institute of Technology, National Defense University, Taipei, Taiwan, in 1980 and the M.S. degree in 1996 from the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, where he is currently working toward the Ph.D. degree with the Department of Computer Science.

His current research interests include information retrieval, web mining, and machine learning.

Chia-Hoang Lee received the Ph.D. degree in computer science from the University of Maryland, College Park, in 1983.

He is currently a Professor with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan. He was a Faculty Member with the University of Maryland and Purdue University, West Lafayette, IN. His current research interests include artificial intelligence, human–machine interface systems, natural-language processing, and opinion mining.

Gen-Chi Lu received the Master's degree in computer science from National Chiao Tung University, Hsinchu, Taiwan, in 2009.

He is currently an Engineer with the Global Legal Division iTEC, Hon Hai Precision Industry Company Ltd., Taipei, Taiwan. His current research interests include natural-language processing, opinion mining, and full-text search.

Emery Jou received the B.S degree in physics from Tsing Hua University, Hsinchu, Taiwan, the M.S. degree in computer science from the University of Texas at Austin, and the Ph.D. degree in computer science from the University of Maryland, College Park.

He is currently a Research Scientist with the Institute for Information Industry, Taipei, Taiwan. He was with several Wall Street firms in the United States for more than 12 years (i.e., Morgan Stanley and JPMorganChase) as a System Architect for Security Transaction Processing through Single Sign-on and Public Key Infrastructure. He was also with Thales nCipher, Cambridge, U.K., where he was engaged in Tape Storage Data Encryption and Key Management Systems. In 2009, he was a Visiting Professor with the College of Computer Science, National Chiao Tung University, Hsinchu. He was also a consultant for the Industrial Technology Research Institute, Hsinchu.