Decision Support Systems 82 (2016) 12–25

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Mining perceptual maps from consumer reviews

Anthony J.T. Lee [a,*], Fu-Chen Yang [a], Chao-Hung Chen [b], Chun-Sheng Wang [c], Chih-Yuan Sun [a]

[a] Department of Information Management, National Taiwan University, Taiwan, ROC
[b] Language Technologies Institute, School of Computer Science, Carnegie Mellon University, USA
[c] Department of Information Management, Jinwen University of Science and Technology, Taiwan, ROC

ARTICLE INFO

Article history:
Received 17 December 2013
Received in revised form 24 September 2015
Accepted 14 November 2015
Available online 3 December 2015

Keywords:
Sentiment analysis
Opinion mining
Latent Dirichlet allocation
Perceptual map
Radar chart

ABSTRACT

Consumer reviews are valuable resources for companies since consumers usually share their using experiences on products or provide useful opinions from various aspects such as different product features. Therefore, in this paper, we propose a method called MPM (mining perceptual map) to automatically build perceptual maps and radar charts from consumer reviews. Perceptual maps and radar charts are business tools widely used in marketing and business analysis. The proposed method may reduce subjective personal bias since perceptual maps and radar charts are mined from a large number of consumer reviews. The analysis results obtained from consumer reviews of smartphones show that the proposed method may provide some practical insights for smartphone companies. Our method can help companies position new products, and formulate effective marketing and competitive strategies.

1. Introduction

Nowadays, online shopping has become a popular way for consumers to buy products. To pick a suitable product from a bunch of choices, consumers may prefer to buy products based on the reviews from other consumers who share their using experiences on the product or provide useful opinions from various aspects such as different product features. Such opinions show how consumers think of the products and in turn reflect their competences [9,14].

Let us consider a review for iPhone 5 from Amazon[1] as shown in Fig. 1. The review of a product may typically include the advantages and disadvantages of the product. For example, in Fig. 1, it is said that iPhone 5 has a bigger screen and a better processor. On the other hand, it has only two product features better than the previous version. This kind of reviews is pretty important and useful for both companies and consumers. For companies, they could know consumers' responses for their products, and what features they have to improve for future products. For consumers, based on this information, they could decide to choose some products to meet their needs.

To efficiently mine useful insights from reviews, many methods have been proposed such as extracting and clustering product features [12,15,17,21,29], and aspect-based opinion mining [6,7,16,23,26]. The aspect-based opinion mining, different from traditional opinion mining which finds overall sentiment from opinions, focuses on how to mine sentiments of different aspects from opinions. However, most of these methods emphasize on improving the efficiency of the existing methods, reducing time complexity in clustering product features and mining aspect-based opinions from reviews. None of them concern with generating valuable insights and business value from companies' perspective.

To gain valuable insights from consumer reviews, we may build a perceptual map to position products developed by a company and its competitors. A perceptual map is a diagram which visually displays the perception of consumers. It is helpful for a company to develop new products or rebrand products since the map clearly shows the positions of products in comparison with those of competitors. For example, Fig. 2(a) illustrates a perceptual map of smartphones. The sentiment in service is a score obtained from the sentiments in consumer reviews about services for each smartphone. Similarly, the sentiment in user experience is a score obtained from the sentiments in consumer reviews about user experiences. iPhone 4 has the highest sentiment score in both service and user experience. Nevertheless, the weakness of perceptual maps is that they could only display some products with respect to two dimensions of product features in a two-dimensional map.

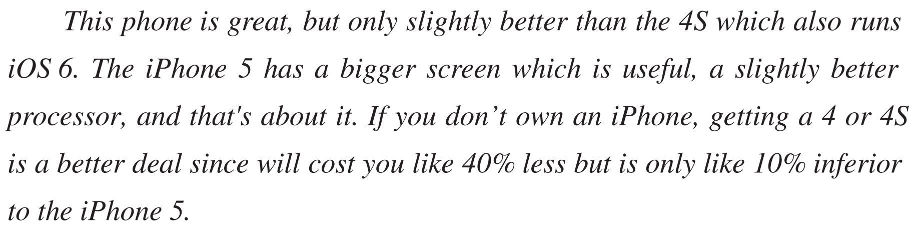
Radar charts could complement some disadvantages of perceptual maps because they could display multiple dimensions of the products in one chart. Nevertheless, the disadvantage of radar charts is that they could only display a limited number of products in a chart. For example, Fig. 2(b) shows a radar chart for HTC Sprint EVO and Samsung Galaxy S, where UX stands for user experience. HTC Sprint EVO performs better in OS, accessory and casing while Samsung Galaxy S performs better in CP value and battery.

Perceptual maps and radar charts are widely used in marketing and business analysis. For example, marketing analysts use them to review

* Corresponding author.
E-mail address: jtlee@ntu.edu.tw (A.J.T. Lee).

[1] http://www.amazon.com.

http://dx.doi.org/10.1016/j.dss.2015.11.002
0167-9236/© 2015 Elsevier B.V. All rights reserved.



This phone is great, but only slightly better than the 4S which also runs iOS 6. The iPhone 5 has a bigger screen which is useful, a slightly better processor, and that's about it. If you don't own an iPhone, getting a 4 or 4S is a better deal since will cost you like 40% less but is only like 10% inferior to the iPhone 5.

Fig. 1. A review for iPhone 5 from Amazon.

the performance of previous positioning strategies and design new ones. Senior managers may use them to gain insights by comparing their products and services with those of their competitors. Also, they may suggest potential entry points in the market. However, as perception is subjective, it is better to ensure that the data to plot the map is unbiased. In practice, the perceptual maps and the radar charts were often made from questionnaires [1,22,24] or by intuitions. If these figures were made from questionnaires, a lot of efforts would be needed to collect enough questionnaires and assure that the questionnaires are unbiased. If they were made by intuitions, the figures might be unreliable because of bias.

Therefore, in this paper, we propose a method called MPM (mining perceptual map) to automatically build perceptual maps and radar charts from consumer reviews. Since the perceptual maps and radar charts are mined from a large number of consumer reviews, MPM can reduce bias in comparison with the methods of building them from questionnaires or by intuitions. The proposed method contains four phases. First, we extract product features from consumer reviews. Second, we create a WordNet-based virtual document for each product feature, where the WordNet-based virtual document of a product feature contains the definition of the product feature in WordNet[2] and the surrounding words that frequently co-occur with the product feature in the same sentence. Third, we modify a latent Dirichlet allocation (LDA) [3], called weighted LDA (WLDA hereafter), and devise a weighted scheme to cluster together similar product features into a feature set by considering both lexical and distributional similarities. Finally, we build perceptual maps and radar charts based on the sentiments on different feature sets. The generated perceptual maps and radar charts are helpful for analysts to formulate effective marketing and competitive strategies.

The results of analyzing consumer reviews of smartphones in both Amazon and PhoneArena datasets from January 2010 to December 2012 show that WLDA achieves the best performance among all comparing methods. Samsung and HTC performed well in processors and operating systems. However, consumers had increasing negative reviews for Apple's operating systems since they expected more dramatic features. In addition, price had a significant influence on sentiment scores in a processor but little influence on sentiment scores in an operating system. Mining perceptual maps and radar charts from a large number of consumer reviews may unveil majority preferences, where the more satisfied consumers are with a feature, the higher sentiment score the feature has. By comparing the experimental results from both datasets, most findings from both datasets are similar to each other. This indicates that MPM is reliable to learn majority preferences of consumers that are helpful for company's decision making.

The contributions of this paper are summarized as follows. First, we construct a virtual document for each product feature based on the definition of the product feature on WordNet and the frequently co-occurred surrounding words of the product feature in consumer reviews. Adding WordNet definitions can enhance the lexical semantics of virtual documents while finding frequently co-occurred surrounding words by a pruning strategy can reduce the effects of noisy words. Thus, the virtual documents can capture the lexical and distributional similarity of product features. Second, we introduce a new weighted scheme and hard constraints in WLDA to help cluster similar product features into a product feature set in which the product features are prone to appear together and share similar lexical meanings. Thus, the clustering performance is improved. Third, we propose the MPM method to automatically build perceptual maps and radar charts from consumer reviews, which may help companies position new products or rebrand products. Finally, we conduct a series of analyses on consumer reviews of smartphones, and find some practical insights from the result analysis.

The rest of this paper is organized as follows. Section 2 surveys the related literature. Section 3 presents the proposed method in detail. Section 4 shows the result analysis. Section 5 summarizes analytical results and discusses how to apply the MPM method to analyze consumer reviews of other products. Finally, the concluding remarks and future work are described in Section 6.

2. Related work

In this section, we review the literature of clustering product features, analyzing sentiment in documents, and building perceptual maps and radar charts.

2.1. Clustering product features

Consumers may describe a product feature in different ways. For example, "ghz" (giga hertz), "quadcore", and "snapdragon" (a family of mobile systems on processors made by Qualcomm) are all product features used to describe "processor". Therefore, it is better to cluster these product features into a product feature set.

To cluster product features together, Liu et al. [17] employed the concept of lexical similarity to cluster similar product features together, where the lexical similarity is defined as the similarity between two terms in semantic networks and thesauri. Many studies [5,10,20] built a semantic network to improve the performance of lexical similarity. By using lexical similarity, two product features are clustered together if the meanings of two product features are close enough. However, some product features are domain-dependent, which have various meanings in different domains. For example, "chips" means potato chips in restaurant reviews; however, it means processor chips in smartphone reviews. Thus, some domain-dependent product features may be misclassified.

On the other hand, some methods [12,18,21,30] use distributional similarities to cluster product features. These methods cluster product features together if they have similar distributions of surrounding words. For example, when people mention the processor of a smartphone, they may describe it by some adjectives (like "fast" and "sluggish") or some nouns (like "ghz", "core", "speed", and "quad"). Therefore, if we mention a product feature only used in a special domain such as "snapdragon", the distribution of surrounding words of "snapdragon" may probably be similar to that of "processor". Thus, "snapdragon" and "processor" may be clustered together.

Matsuo et al. [18] applied the concept of distributional similarity to merge terms together if they have similar distributions of surrounding words, where distributional similarity is defined as the similarity between the occurrences of surrounding words of both terms. Guo et al.

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

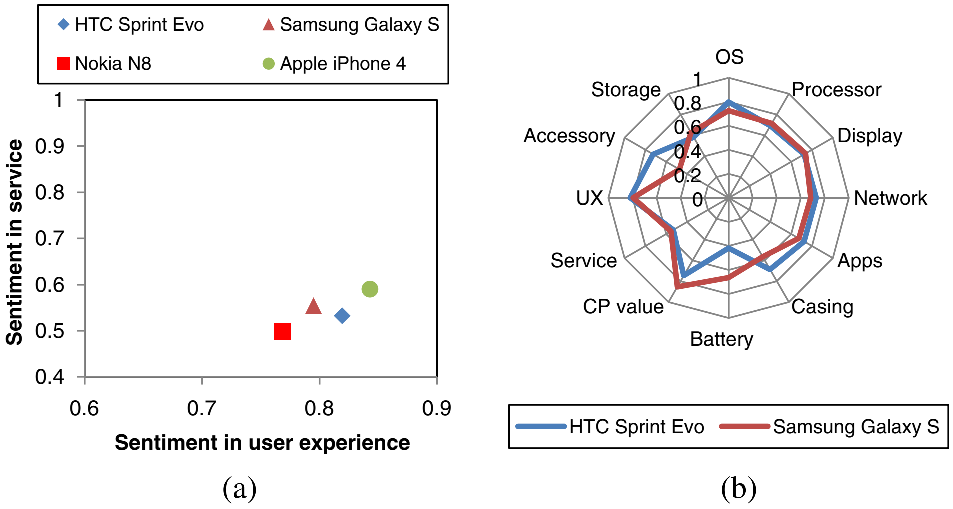


Fig. 2. Perceptual map and radar chart of smartphones.

[12] proposed a two-level latent semantic association (LaSA) model to cluster product features, where LaSA also uses distributional similarity for clustering. Zhai et al. [30] developed a constraint-based latent Dirichlet allocation (CBLDA), where must-links and cannot-links are set between product features. A must-link constraint means that two product features must be in the same cluster while a cannot-link constraint means that two product features cannot be in the same cluster. However, these constraints can be relaxed in the sampling process. Since LDA [3] is widely adopted as variants in many applications and CBLDA [30] is the state-of-the-art method in clustering product features, we consider LDA and CBLDA as comparing methods in our evaluation. Note that these methods do not consider the meanings of product features, and they may misclassify product features.

To analyze consumer reviews of a specific category of products such as smartphones, we first construct the WordNet-based virtual document of a product feature by using both the definition of the product feature in WordNet and the surrounding words of the product feature in consumer reviews. That is, we utilize both distributional and lexical similarities in our method to construct virtual documents. Next, we propose a weighted LDA (WLDA) to cluster similar product features into a feature set. By taking the degree of co-occurrences between words into account, WLDA selects the surrounding words statistically relevant to the product feature to reduce the impacts caused by irrelevant words, and gives different weights to various product features to improve the clustering performance. Finally, unlike CBLDA in which two similar product features imposed by a must-link (soft-link) constraint may be possibly clustered into different feature sets, WLDA enforces some frequently-used product features to be clustered into the predefined feature sets since these product features of a specific category of products can be well grouped together by experts.

2.2. Analyzing sentiment in documents

Many sentiment analysis methods focus on detecting user's sentiments and opinions in sentence level [28] or document level [2,27]. However, a consumer review may include both positive and negative opinions toward different aspects. For example, a consumer may claim that the quality of the product is excellent but complain about the consumer service of the company in a review. Consequently, these methods cannot discover consumers' sentiments in different aspects.

Therefore, Hu and Liu [15] proposed the concept of aspect-based opinion mining to find the sentiments in different features from reviews. Titov and McDonald [23] used a statistical model to discover topics in documents and extract textual evidence supporting the rating of each topic. Carrillo de Albornoz et al. [6,7] predicted the rating of hostels based on consumer opinions toward different aspects. Wang et al. [26] presented a probabilistic regression model to discover the latent opinion on each aspect for each reviewer. Based on the concept of aspect-based opinion mining, our method extracts consumer sentiments in different aspects. Moreover, we devise a pruning strategy to remove irrelevant words and a weighted scheme to improve the clustering performance.

2.3. Building perceptual maps and radar charts

Perceptual maps are often used to discover insightful strategies. Bhatnagar and Ghose [1] used perceptual maps to display the result of segmenting web shoppers' behaviors. Vanlaar et al. [25] employed a perceptual map to explain the public concerns about dangerous driving behaviors. Bose and Gupta [4] utilized a perceptual map to present the experimental results based on ratings given by customers of three public sector banks and three new generation private banks. All of these methods use a perceptual map to show the result obtained from questionnaires.

Netzer et al. [19] presented a method to analyze market-structure surveillance and build a perceptual map to show the result. Since most perceptual maps use two attributes to present the results, Green [11] proposed a multi-dimensional scaling method to resolve such a problem and visualize multiple attributes in a two-dimensional graph.

Radar charts were first used by Georg von Mayr in 1877.[3] It is a useful way to present data in multiple dimensions. Han and Huang [13] used radar charts to display risk patterns in different levels of project performance. Chang et al. [8] utilized radar charts to visualize the many-to-one relationships between QoS (Quality of Service) and QoE (Quality of Experience). In this paper, we use radar charts to complement the disadvantage of perceptual maps since radar charts can display multiple product features simultaneously. The limitation of radar charts is that they may display fewer products than perceptual maps.

In comparison with the methods of building perceptual maps and radar charts from questionnaires, our proposed method can reduce bias since the perceptual maps and radar charts are mined from a large number of consumer reviews.

[3] http://en.wikipedia.org/wiki/Radar_chart.

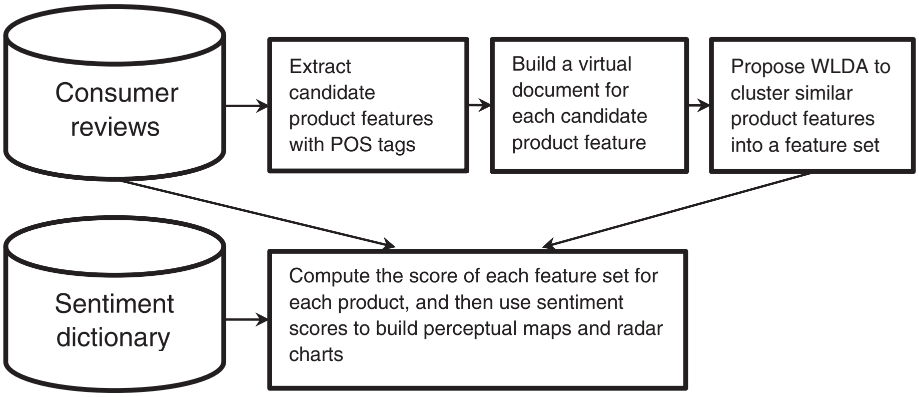

Consumer reviews
Extract candidate product features with POS tags
Build a virtual document for each candidate product feature
Propose WLDA to cluster similar product features into a feature set
Sentiment dictionary
Compute the score of each feature set for each product, and then use sentiment scores to build perceptual maps and radar charts

Fig. 3. The framework of the MPM method.

3. The proposed method

In this section, we propose a method, called mining perceptual map (MPM), to automatically build perceptual maps and radar charts from consumer reviews. The framework of MPM is shown in Fig. 3. MPM contains four phases. First, we use part-of-speech tagging[4] to mark part of speech for each term. Like the previous study [15], we remove the terms with high frequencies (stop words), and only consider nouns and noun phrases as candidate product features. Second, after extracting candidate product features, we build a virtual document for each candidate product feature, where the virtual document contains the definition of the candidate product feature in WordNet and the surrounding words which frequently co-occur with the candidate product feature in the same sentence. Third, we design a weighted latent Dirichlet allocation (WLDA) to cluster similar candidate product features into a feature set. Finally, for each feature set of each product, we compute its sentiment score by the positive and negative sentiments in the sentences containing the candidate product features in the feature set. Then, we build perceptual maps and radar charts based on the computed sentiment scores.

3.1. Extracting candidate product features

We first use Stanford POS Tagger[5] to tag the part-of-speech of each word in consumer reviews. According to previous studies [12,15,21], most of the product features are nouns such as screen, resolution, and battery. Thus, we only consider nouns and noun phrases as candidate product features. For example, we extract "iOS", "screen", "processor", and "deal" as candidate product features from the consumer review shown in Fig. 1.

3.2. Building a virtual document for each candidate product feature

The input documents to WLDA are different from those used in the original LDA [3]. We first use the concept of distributional similarity [12,30] to construct virtual documents. For each candidate product feature *f*, we extract all the surrounding words of *f* in the consumer reviews to form a virtual document, where the surrounding words are the words appearing in the same sentence as *f*. However, some surrounding words may be irrelevant to *f*. Thus, we devise a pruning strategy to remove irrelevant surrounding words. A surrounding word is relevant to *f* if they frequently co-occur in a sentence. We introduce the Jaccard coefficient[6] to measure the co-occurrence (or relevance) between two words, *A* and *B*, as shown in Eq. (1), where S_A contains the sentences including *A*, S_B contains the sentences including *B*, $|S_A∩S_B|$ and $|S_A∪S_B|$ which are the sizes of the intersection and union of S_A and S_B, respectively. If $J(A,B)$ is not less than a predefined threshold θ, *A* and *B* are relevant; otherwise, they are not.

$$J(A,B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|}. \tag{1}$$

Consider the consumer review shown in Fig. 1, where "processor" is extracted as a candidate product feature. If *J*("better", "processor") is not less than θ, "better" is relevant to "processor" and added to the virtual document of "processor".

However, it may be difficult to cluster similar candidate product features with different surrounding words into a feature set if we do not consider their meanings. Therefore, we use WordNet to construct virtual documents and resolve such a problem. WordNet is one of the largest lexical databases for English developed by Princeton University. It contains the synsets and definitions of words. For each candidate product feature, if it is defined in WordNet, we add the words in the definition to its virtual document. Thus, we construct a WordNet-based virtual document of a candidate product feature by using both the definition words in WordNet and the relevant surrounding words of the candidate product feature. That is, we utilize both distributional and lexical similarities in our method to construct virtual documents. However, the definition of a candidate product feature in WordNet usually contains twenty to forty words. In comparison with a virtual document containing thousands of surrounding words, we have to add more definition words to improve the performance. Thus, the WordNet-based virtual documents are constructed in the following two steps.

a. For each candidate product feature in WordNet, add the definition of the candidate product feature to the virtual document so that the percentage of the definition words in the virtual document is not less than a predefined threshold η. We may repeat the definition words several times until the percentage of these words meets the requirement.
b. To improve the efficiency of WLDA, we restore inflected words to their stems. We also change the frequency of each word into the logarithm of its frequency to reduce the number of words in the virtual document. Taking the logarithm can retain the property of the virtual document because logarithm is a monotonically increasing function. In the experiment, we apply $\text{floor}(\log_2(q)) + 1$ to each word, where q is the frequency of the word in the virtual document. That is, we change the frequency of the word from q into $\text{floor}(\log_2(q)) + 1$.

Let us consider the example in Fig. 1 again and assume that *J*("better", "processor") > θ. Since "better" is relevant to "processor", it is added to the virtual document of "processor". In addition, the definition of "processor" in WordNet contains "processor", "central_processing_unit",

[4] http://en.wikipedia.org/wiki/Part-of-speech_tagging.
[5] http://nlp.stanford.edu/software/tagger.shtml.
[6] http://en.wikipedia.org/wiki/Jaccard_index.

"CPU", "mainframe", "part", "computer", "microprocessor", "chip" and "data", where the stop words are excluded. These words are added to the virtual document too. Thus, the virtual document of "processor" is $\text{Vdoc}_{\text{processor}}$ = {better, processor, central_processing_unit, CPU, mainframe, part, computer, microprocessor, chip, data}. In this example, just one surrounding word is added to the virtual document since the example just contains a consumer review. When many consumer reviews are considered, the virtual document may contain thousands of surrounding words. Therefore, we need to perform steps (a) and (b) to balance the amount of surrounding words and definition words, and reduce the number of words in the virtual document.

3.3. Clustering product features

To improve the performance of LDA, we introduce a weighted scheme to LDA and propose a weighted LDA (WLDA). In WLDA, we first predefine some product features in each product feature set. Next, based on the predefined product features, we devise a weighted scheme to cluster similar candidate product features into a feature set, where the weighted scheme is used to compute the similarity (or relevance) between a word and predefined product features. WLDA is a probabilistic model used to discover latent topics (feature sets) in consumer reviews, where each consumer review may contain multiple feature sets and each feature set may contain multiple words.

We use the Gibbs sampling algorithm[7] to implement WLDA. For each Gibbs sampling iteration, we assign each word in virtual documents to a feature set according to the probability that the word is assigned to each feature set. To compute the probability, for each word w_i in virtual documents, we check if w_i belongs to any predefined feature set. If this is the case, w_i is directly assigned to the predefined feature set. Otherwise, we use Eq. (2) to compute the probability that w_i is assigned to a feature set k,

$$P(z_i = k|W, z_{-i}) = \frac{WF_{w_i,k} + \beta}{\sum_j WFw_j, k + V\beta} * \frac{FD_{k,d} + \alpha}{\sum_l FD_{l,d} + K\alpha} * sim(w_i, k) \quad (2)$$

where z_i is a candidate feature set of w_i, W contains all the words in virtual documents, z_{-i} contains all the feature sets except z_i, $WF_{wi,k}$ denotes how many times w_i is assigned to k, $FD_{k,d}$ denotes how many words in virtual document d are assigned to k, V is the number of distinct words in virtual documents, K is the number of feature sets, $sim(w_i,k) = \frac{1}{N}\sum_{l=1}^{N} J(w_i, w_l)$ is the similarity between w_i and k, w_l is a predefined product feature in k, N is the number of predefined product features in k, and α and β are the smoothing parameters used to decide Dirichlet distributions, respectively. If there are no predefined product features for a certain feature set k, the top N words ranked by the words' probabilities in $WF_{wi,k}$ are used.

To implement the Gibbs sampling algorithm, we first randomly assign each word to a feature set. Next, for each iteration of Gibbs sampling, we compute the probability distribution over feature sets for each word. During the sampling process, if any feature set assignment is changed, we update the matrices *WF* and *FD* accordingly. The previous steps are performed repeatedly until the number of iterations is reached the predefined threshold or the updates of both *WF* and *FD* are converged.

After WLDA finishes, we obtain a probability distribution over feature sets for each word. By referring to the *WF* matrix, for each word, we can find the feature set that the word appears most frequently, and label the word as the feature set. Next, for each candidate product feature *f*, we assign it to the feature set containing the majority of words in *f*'s virtual document. If there are two or more feature sets with the same number of words, we use the Jaccard coefficient to compute the relevance between *f* and each candidate product feature in these feature sets, and then take the average of the computed Jaccard coefficients for each feature set. *f* is assigned to the feature set with the largest average Jaccard coefficient.

Let us consider the virtual documents after WLDA finishes as shown in Fig. 4, where the superscript *k* (label) of a word denotes that the word appears most frequently in the *k*th feature set. Based on these labels, for each virtual document, we can compute how many words in the virtual document are in each feature set. For example, we find that 80% of words in Android's virtual document ($\text{Vdoc}_{\text{Android}}$) are in feature set 2. Thus, "Android" is assigned to feature set 2. Similarly, "iOS" is assigned to feature set 2 since 70% of words in iOS's virtual document are in feature set 2. Likewise, "Snapdragon" and "processor" are assigned to feature set 1. Finally, we cluster similar product features into the same feature set. By observing the product features in the feature sets, we may find that most product features in feature sets 1 and 2 are related to processor and operating system, respectively.

The differences between WLDA and CBLDA [30] can be elaborated in three aspects. First, CBLDA uses soft-constraints; however, WLDA uses hard constraints. To analyze the consumer reviews of a specific category of products, some frequently-used product features can be well grouped together into a feature set by experts. Thus, WLDA always clusters the product features in a constraint into the same feature set. However, CBLDA allows the product features in a constraint to be clustered into different feature sets. This may possibly lead to misalignment between frequently-used product features and feature sets, and result in poorer cluster performance. For example, we may know the product features "processor" and "chip" should be clustered together in smartphone consumer reviews. CBLDA would only increase the tendency to be clustered together but still possibly cluster the two product features into different feature sets. However, both product features are guaranteed to be clustered into the same feature set by hard constraints. Thus, WLDA can well cluster relevant product features together. Second, we consider both lexical and distributional similarities in WLDA when clustering product features into feature sets. For distributional similarity measurement, only relevant surrounding words are added to virtual documents. Meanwhile, we further take the semantic meanings in account by including the definition words in WordNet into virtual documents. Because CBLDA only uses all non-stop words in consumer reviews to generate virtual documents, it may suffer from noisy words and ignoring the semantic similarity between product features. Third, we introduce a weighted scheme ($sim(w_i,k)$) in the sampling process. If a word is more relevant to the predefined features of product feature *k*, the word is more likely to be assigned to *k*. Although CBLDA employs must-links and cannot-links to generate a weight as well, the constraints used in CBLDA may not well cluster product features. For example, in CBLDA, the cannot-link constraint specifies that if two product features are in the same sentence but not connected by "and", they form a cannot-link. However, in the sentence "The front facing camera is 1.3 MP", both product features "front facing camera" and "MP" are not connected by "and" but they should belong to the same feature set, where MP stands for megapixel. In addition, the must-link constraint specifies that if two product features share one or more words, they form a must-link. However, the product features "storage capacity" and "battery capacity" should belong to different feature sets. Therefore, the clustering performance of WLDA is better than that of CBLDA.

3.4. Building perceptual maps and radar charts

After clustering candidate product features into feature sets, we use resultant feature sets to build perceptual maps and radar charts. Each feature set is considered as a dimension in the map.

For each product, we divide all reviews of the product into sentences. For each sentence containing a candidate product feature, the positive or negative sentiment count of *fs* is incremented by 1 according to the polarity (positive or negative) of the closest sentiment word if the sentence contains any sentiment words, where *fs* is the

[7] http://en.wikipedia.org/wiki/Gibbs_sampling.

$\text{Vdoc}_{\text{Android}}$={os^2, memory^3, core^1, windows^2, release^2, version^2, operating^2, system^2, cream^2,run^2}

Vdoc_{iOS}={operating^2, system^2, develop^2, core^1, work^4, release^2, version^2, system^2, platform^2, ram^3}

$\text{Vdoc}_{\text{snapdragon}}$={chip^1, quad^1, core^1, chip^1, storage^3, dualcore^1, qualcomm^1, ghz^1}

$\text{Vdoc}_{\text{processor}}$={core^1, core^1, qualcomm^1, core^1, chip^1, quadcore^1, memory^3, ghz^1, ghz^1}

Fig. 4. Five virtual documents after WLDA finishes.

feature set containing the candidate product feature. The above steps are performed sentence by sentence. Finally, we obtain the positive and negative sentiment counts of each feature set of the product.

Let us consider the statement shown in Fig. 1, “The iPhone 5 has a bigger screen which is useful, a slightly better processor”. In the sentence, “screen” is selected as a candidate product feature, which is clustered into the feature set “display”. The sentence also contains a positive sentiment word “bigger”, which is the closest sentiment word to “screen”. Thus, the positive sentiment count of “display” is incremented by 1. Similarly, “processor” is selected as a candidate product feature, which is clustered into the feature set “processor”. The sentence also contains a positive sentiment word “better”, which is the closest sentiment word to “processor”. Thus, the positive sentiment count of “processor” is incremented by 1.

Next, we use the positive and negative sentiment counts to compute the score of each feature set of a product as shown in Eq. (3), where $pos_j(k)$ stands for the positive sentiment count of feature set k for product j and $neg_j(k)$ stands for the negative sentiment count of feature set k for product j.

$$SC_j(k) = \frac{pos_j(k)}{pos_j(k) + neg_{j(k)}}. \tag{3}$$

To build a perceptual map, we first select some products and two feature sets. Next, we use the sentiment scores of these two feature sets for the selected products to draw a perceptual map. Then, we can analyze the strength and weakness of each product on the perceptual map. Similarly, to build a radar chart, we first select a few of products and some (or all) feature sets. Next, we use the sentiment scores of the selected feature sets for the selected products to draw a radar chart, and then analyze the strength and weakness of each product on the radar chart. If many products are shown in a radar chart, the chart contains too many overlapping line segments so that the chart may not be easy to read. Thus, it may be required to limit the number of products shown on a radar chart. Since a perceptual map only shows two feature sets at a time, we use radar charts to complement the disadvantage of perceptual maps, where multiple feature sets can be simultaneously displayed on a radar chart.

3.5. The MPM method

The pseudo code the MPM method is shown in Fig. 5. In step 1, all nouns and noun phrases from each consumer review are extracted as candidate product features. For each candidate product feature f, in steps 3–7, we scan the consumer reviews sentence by sentence to find f's surrounding words. Once a surrounding word is found, we check if it is relevant to f by Eq. (1). If this is the case, the surrounding word is added to f's virtual document. Also, in steps 8–10, we check if f is defined in WordNet and add the words in the definition to f's virtual document. Thus, the virtual document of f may contain the relevant surrounding words in consumer reviews and the definition words in WordNet.

In steps 12–29, we apply the Gibbs sampling algorithm to implement WLDA. First, we randomly assign each word to a feature set and then update the matrices WF and FD according to the random assignments in steps 12–13, where $WF_{wi,k}$ records how many times word w_i is assigned to feature set k, and $FD_{k,d}$ records how many words in virtual document d are assigned to k. For each Gibbs sampling iteration, we keep refining the matrices WF and FD in steps 14–28 by excluding the current feature set assignment of w_i. To exclude the current assignment, we set $WF_{wi,k'} = WF_{wi,k'} - 1$ and $FD_{k',d} = FD_{k',d} - 1$ in step 16, where k' is the current feature set assignment of w_i and d is the virtual document containing w_i. This step ensures that the refinement is not influenced by the current assignment. Next, we check if w_i is a predefined product feature. If this is the case, w_i is assigned to the predefined feature set in step 17. Otherwise, we compute the probability that w_i is assigned to each feature set by excluding the current feature set assignment and accumulate these probabilities to vector G by feature set in steps 21–23, where G records the accumulated probability that w_i is assigned to each feature set. Then, we assign w_i to a certain feature set according to the probability distribution of feature set assignments recorded in G in step 24, where the probability that w_i is assigned to feature set k is proportional to G_k, the kth element of G. After performing the feature set assignment of w_i, we update the matrices WF and FD by setting $WF_{wi,k''} = WF_{wi,k''} + 1$ and $FD_{k'',d} = FD_{k'',d} + 1$ in step 26, where k'' is the new feature set assignment of w_i. After the Gibbs sampling process is finished, we obtain a probability distribution over feature sets for each word, where the Gibbs sampling process is finished when the number of iterations is reached the predefined threshold, or the updates of both WF and FD are converged. By referring to the WF matrix, for each word, we can find the feature set that the word appears most frequently, and label the word as the feature set. Then, we assign each candidate product feature to the feature set containing the majority of words in its virtual document in step 29. As a result, a feature set contains a set of similar product features. Finally, we compute the sentiment score of each feature set for each product by Eq. (3) in steps 30–35, and then build the perceptual maps and radar charts based on the sentiment scores computed in step 36.

4. Result analysis

As the global smartphone market has grown quickly in recent years, we evaluate our method using the consumer reviews of smartphones. We first introduce the datasets in Subsection 4.1. Next, we evaluate the clustering performance of the proposed method in Subsection 4.2. Finally, we present the analytical results in Subsections 4.3 and 4.4.

4.1. Datasets

According to the survey from IDC Worldwide Mobile Phone Tracker[8] on January 24, 2013, the top four smartphone vendors in 2012 were Samsung, Apple, Nokia, and HTC. Thus, we used the reviews of smartphones made by these four vendors as our datasets during the period from January 2010 to December 2012.

We collected the reviews of smartphones from Amazon[9] and PhoneArena.[10] The number of products (smartphones) and number of

[8] https://www.idc.com/getdoc.jsp?containerId=prUS23916413.

[9] http://www.amazon.com/.

[10] http://www.phonearena.com/.

```
Method: MPM
Input :   A collection of consumer reviews $R$ and the number of feature sets $K$, where each
          feature set contains $m$ predefined product features
Output :  Perceptual maps and radar charts
    // Phase 1: Extract candidate product features with POS tags
1.  Extract nouns and noun phrases (candidate product features) from each consumer
    review in $R$ and collect the extracted candidate product features into $C$;
    // Phase 2: Build a virtual document for each candidate product feature
2.  foreach candidate product feature $f$ in $C$ do
3.      foreach surrounding word $w$ of $f$ do
4.          if $J(w,f)$ is not less than $\theta$ then
5.              Add $w$ to $f$'s virtual document;
6.          end if
7.      end for
8.      if $f$ is defined in WordNet then
9.          Add the words in the definition to $f$'s virtual document and repeat these words so
            that the percentage of added words in the virtual document is not less than a
            predefined threshold $\eta$;
10.     end if
11. end for
    // Phase 3: Use WLDA to cluster similar product features into a feature set
12. For each word in virtual documents, randomly assign it to a feature set;
13. Update $WF$ and $FD$ according to the random feature set assignments in step 12;
14. while Gibbs sampling process is not finished do
15.     foreach word $w_i$ in virtual documents do
16.         Exclude the current feature set assignment of $w_i$ by setting $WF_{w_i,k'} = WF_{w_i,k'}-1$ and
            $FD_{k',d} = FD_{k',d}-1$;
17.         if $w_i$ is a predefined feature then
18.             $w_i$ is assigned to the predefined feature set, denoted by $k''$;
19.         else
20.             Initialize a $K$-dimensional vector $G$, recording the accumulated probability of
                each feature set assignment;
21.             foreach feature set $k$ do
22.                 Compute $P(z_i=k|W,z_{-i})$ by Eq. (2) and add it to $G_k$, where $G_k$ is the $k$th
                    element of $G$;
23.             end for
24.             Sample a feature set $k''$ according to the probability distribution of feature set
                assignments recorded in $G$, and assign $w_i$ to the feature set $k''$;
25.         end if
26.         Update $WF$ and $FD$ by setting $WF_{w_i,k''}=WF_{w_i,k''}+1$ and $FD_{k'',d} = FD_{k'',d}+1$;
27.     end for
28. end while
29. Assign each candidate product feature to the feature set containing the majority of
    words in its virtual document;
    // Phase 4: Compute the sentiment score of each feature set for each product, and build
    perceptual maps and radar charts based on the sentiment score computed
30. foreach product $j$ do
31.     Scan all consumer reviews of product $j$ sentence by sentence to compute the positive
        and negative sentiment counts for each feature set;
32      foreach feature set $k$ do
33.         Compute the sentiment score $SC_j(k)$ by Eq. (3);
34.     end for
35. end for
36. Use the sentiment scores computed to build perceptual maps and radar charts;
37. end
```

Fig. 5. The pseudo code of the MPM method.

Table 1
Consumer reviews.

Brand	Number of products		Number of reviews		Number of sentences	
	Amazon	PhoneArena	Amazon	PhoneArena	Amazon	PhoneArena
HTC	53	65	4482	678	60,323	31,281
Samsung	114	122	10,980	1109	116,270	58,922
Apple	3	3	1323	157	8494	11,733
Nokia	42	40	4941	449	60,810	27,431
Total	212	230	21,726	2393	245,897	129,367

reviews of these four vendors are listed in Table 1. According to the survey from IDC Worldwide Mobile Phone Tracker, Samsung became the largest vendor of smartphones in June 2011. Thus, Samsung has had most reviews on Amazon and PhoneArena since it became the largest smartphone provider.

4.2. Performance of clustering product features in Amazon dataset

To obtain ground truth labels, we employ three experts to label the product features extracted from consumer reviews. The total number of reviews is 21,726 as listed in Table 1. From these reviews, we extract all nouns and noun phrases as candidate product features. There are 35,484 candidate product features in total. Next, we remove the infrequent candidate product features which appear less than 20 times in all reviews. Three experts tag these candidate product features into 13 feature sets, namely, OS, processor, display, network, application (app), casing, battery, price, service, user experience, accessory, storage, and unrelated. If a candidate product feature is not related to any product feature listed, it is labeled as unrelated. If more than two experts tag a product feature in the same feature set, we designate the product feature into the feature set. Finally, there are 413 product features tagged into the first 12 feature sets by the experts. For each feature set, we predefine 3 product features, which are randomly picked from the labeled feature sets.

Rand index [30] has been widely used to evaluate the performance of clustering product features. Rand index is defined by Eq. (4), where a denotes the number of pairs of product features clustered into the same cluster and also tagged in the same cluster by the ground true labeling, b denotes the number of pairs of product features clustered into the different clusters and also tagged in the different clusters by the ground true labeling, and m denotes the number of product features in total. The larger the Rand index is, the better the clustering result is.

$$\text{Rand index} = \frac{2(a+b)}{m(m-1)}. \tag{4}$$

To determine the best combination of parameters to be used in the experiments, we randomly sample a 30% dataset as the training dataset, run some experiments and find that the best combination of parameters is in the region where $12 \le K \le 22$, $0 \le \theta \le 0.05$, and $0 \le \eta \le 50\%$. Then we perform a grid search method on all combinations of parameters, where the number of feature sets K ranges from 12 to 22 by step of 2, Jaccard coefficient threshold θ from 0.01 to 0.05 by step of 0.01 and WordNet percentage threshold η from 10% to 50% by step of 10%. We compute the Rand index for each combination and then choose the best one. As a result, the best Rand index is obtained when $K = 14$, $\theta = 0.04$ and $\eta = 20\%$. Fig. 6 illustrates the Rand indices by varying θ and η, where K is set to 14.

Next, we compare WLDA with the original LDA [3], CBLDA [30], VLDA, PLDA, WLDA-J and WLDA-W. The original LDA is the baseline for our experiments. CBLDA is a soft-link method using must-links and cannot-links. VLDA is the original LDA implemented by using WordNet-based virtual documents. PLDA is the original LDA with 3 predefined product features for each of 12 feature sets. These predefined product features are the same as those used in WLDA. WLDA-J is WLDA without applying the weighted scheme while WLDA-W is WLDA without using the pruning strategy. For each method, we determine the best combination of parameters for the method in the same way as that for WLDA. As a variant of WLDA, the best parameter combination of WLDA-J is the same as that of WLDA. Since WLDA-W and VLDA do not adopt the pruning strategy, they do not use parameter θ. Their best combinations of parameters are $\eta = 10\%$ and $\eta = 20\%$, respectively. Like the original LDA, PLDA and CBLDA have just one parameter, the number of feature sets. The best number of feature sets for each method is determined by Fig. 7.

Fig. 7 shows Rand index versus the number of feature sets. CBLDA outperforms LDA, VLDA and PLDA in every case because CBLDA incorporates soft-links to group similar product features together. WLDA, WLDA-J and WLDA-W perform better than CBLDA since they take the advantage of hard-links, lexical and distributional similarities. WLDA has the best performance among those methods since it combines both the weighted scheme and the pruning strategy. The performance improvement will in turn contribute to generating more accurate perceptional maps and radar charts. This is because high Rand index indicates that WLDA is able to cluster similar product features into a feature set, which is useful in well positioning products and quickly gaining business insights.

Fig. 8 illustrates the product features in each feature set. The feature set in the first column of the first row contains product features “micro”, “sd”, and “gb”. All of them are related to “storage”. The feature set in the second column of the first row contains product features “sandwich” and “ice cream”. These product features are not related to food, but are related to “operating system” since “ice cream sandwich” is the version 4.0 of the Android operating system for smartphones.

4.3. Results of Amazon dataset

4.3.1. Perceptual maps by brand

We first build perceptual maps derived from MPM by brand for each year from 2010 to 2012. We first pick the most popular products from each brand, each of which has the largest number of reviews in the time period. Fig. 9 illustrates the perceptual map of operating system and processor. For operating system, the iOS of Apple had good performance in the first period; however, its sentiment score decreased a lot

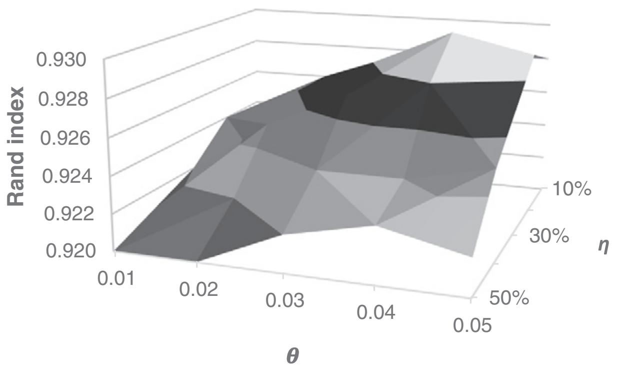


Fig. 6. Rand indices for different combinations of thresholds.

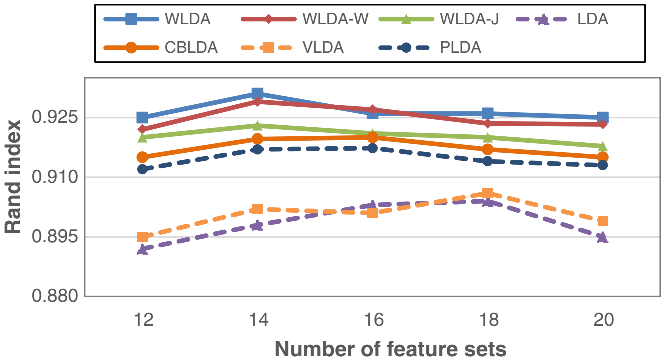


Fig. 7. Rand index versus number of feature sets.

in 2012 since consumers had more and more negative opinions. This is because consumers felt disappointed that there were not many dramatic new features added to the new iOS while holding extremely high expectations on iPhone 5. Also, many of the new features were similar to those that Android phones had had. On the contrary, the OS of Samsung and HTC appeared to receive more and more positive opinions where they were both Android-based systems. As new Android versions had been consistently updated from version 2.1 (Eclair), 2.2 (Froyo), 2.3 (Gingerbread), 3.0–3.2 (Honeycomb), 4.0–4.0.4 (Ice cream sandwich) to 4.1–4.2 (Jelly bean), consumers may enjoy the advantages of the updates in each period. Especially, HTC developed its own user interface, called HTC SENSE, which gradually received more compliments. Similarly, Nokia replaced Symbian OS by Windows Phone 8 in which the system was flesh and friendly to customers. This may indicate that consistently keeping customers aware of adding new and surprising features to OS is an important strategy. For processor, Apple had almost the stable sentiment score in three years; however, Samsung improved the sentiment scores significantly. Since Samsung Galaxy S incorporated a quad-core processor while the smartphones of the other 3 companies still used dual-core processors, Samsung satisfied more consumers' requirements in which they needed more computing power in their apps.

Next, we compare some results obtained by WLDA and those by CBLDA since CBLDA has the best performance among the previously proposed methods. Fig. 10 illustrates the perceptual maps of operating system and processor generated by CBLDA. The perceptual maps of CBLDA show two opposite trends in comparison with those of WLDA. First, the perceptual maps generated by CBLDA indicate that iPhone's sentiment of processor declines gradually year by year while it slightly increases in those maps generated by WLDA. According to the benchmark of a third party,[11] the processor of iPhone 5 (A6) runs about 2.5 times faster than that of Phone 4S (A5), which is similar to Apple's claims. Although consumers might not experience exactly as fast as the benchmark presented, there was not a significant increase of the complaints about iPhone's computing power. In fact, iPhone consumers were satisfied with the new processor since many of them mentioned the smoothness when running applications of iOS. Second, CBLDA shows a declining sentiment score of operating system for Samsung smartphones while WLDA presents a growing sentiment score. Many statistics showed that Android (OS adopted by Samsung) smartphones became popular in terms of user satisfaction[12] and market share[13] in comparison with iOS. Many consumers were using Android smartphones due to several reasons[14] such as free OS (lower price), numerous apps and customization, and adoption by popular brands. Table 1 also shows the considerable rise in the number of Samsung's customer reviews.

We may find that CBLDA may make inappropriate positioning of product features. This is because CBLDA may not be able to cluster similar product features into a feature set. If the product features are not well clustered, we may not discover the growth of sentiment of Samsung OS and iPhone processor. Such misleading trends may prevent companies from understanding the genuine feedback of consumer reviews and discovering business insights.

Fig. 11 presents the radar charts for the products of Samsung and HTC from 2010 to 2012. A radar chart can display multiple feature sets at the same time when we focus on a fewer products. In 2011, Samsung Galaxy S II dominated HTC Inspire in almost all feature sets, except for operating system since they were both Android-based systems. Consequently, in 2012, HTC improved its product and released a new smartphone, HTC One V, which was more competitive in most feature sets.

4.3.2. Perceptual maps by price

In this section, we build perceptual maps by price. We divide the products into two groups by price, where the products of low-price (less than $300) are marked by hollow patterns and the products of high-price (not less than $300) are marked by solid patterns. Then, we pick the most popular product of each brand in each price range, except Apple, since the price of every product of Apple is greater than $300.

Fig. 12 shows that price has a significant influence on sentiment scores in processor. Consumers tend to have more positive sentiments on high-price products. Although the processor of Apple did not have a good performance, generally speaking, the sentiment scores in processor of high-price products, except Apple, continued to increase year by year. However, price does not have the same effect on sentiment scores in operating system since the low-price products still use a stable operating system, such as Android or Windows. There is little difference in operating systems between high-price and low-price products.

4.4. Results of PhoneArena dataset

After performing the grid search method, we find that the best parameter combination of WLDA is $K = 14$, $\theta = 0.05$ and $\eta = 20\%$ for the PhoneArena dataset. By comparing the results obtained from the PhoneArena dataset with those from the Amazon dataset, we find that most results are similar to each other although the sentiment scores may be different between both datasets. For example, Fig. 13 shows the perceptual map of operating system and processor obtained from the PhoneArena dataset. For operating system, the sentiment score in iOS of Apple decreased continuously since consumers had more and more negative opinions. On the other hand, the OS of Samsung and HTC appeared to receive more and more positive opinions since consumers were getting more satisfied with the Android system. For processor, Apple kept the sentiment scores stably in three years; however, Samsung improved its sentiment scores significantly.

5. Discussion

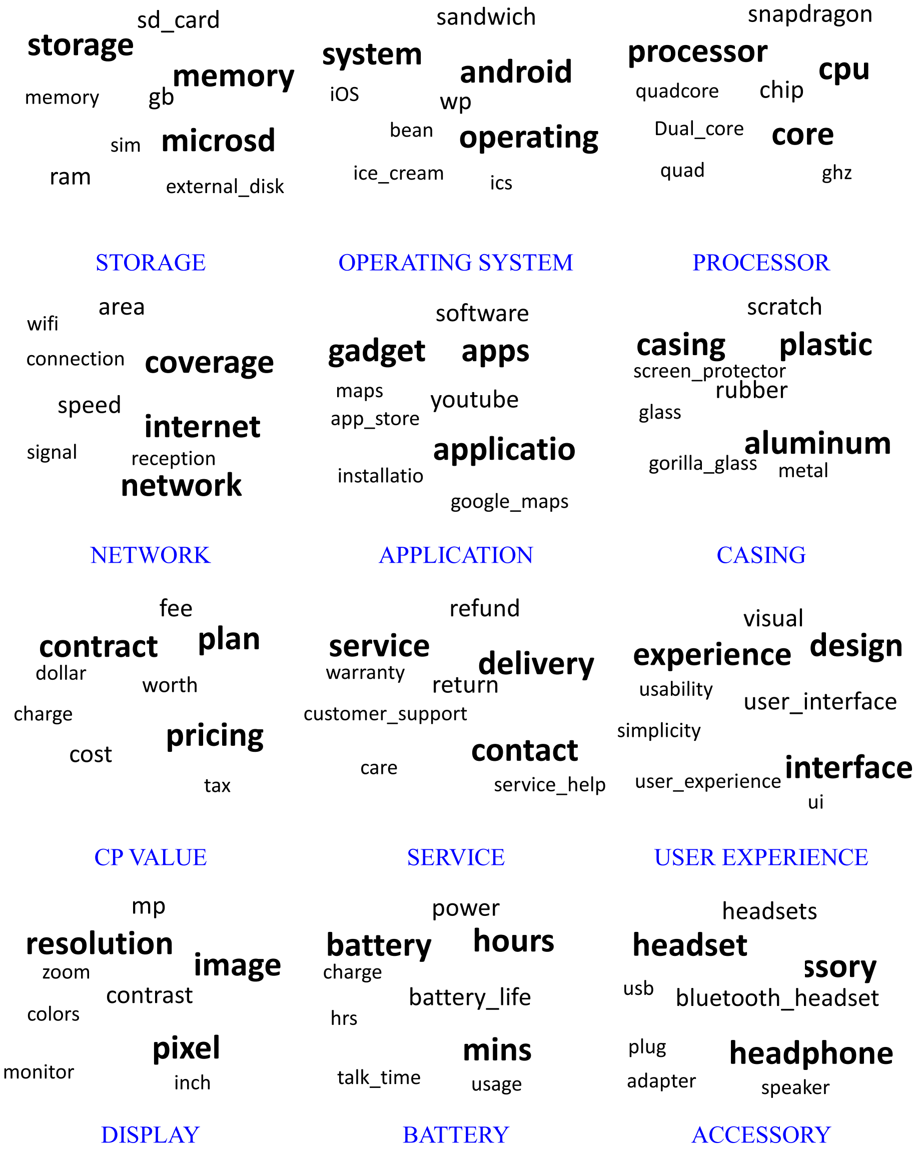
It has been shown that the MPM method can automatically build perceptual maps and radar charts from a large number of smartphone reviews in both Amazon and PhoneArena datasets. The resultant maps or charts may help gain insights into business initiatives in reviewing the performance of their products with those of their competitors. As Amazon is one of the largest online retailers in the world, it has attracted numerous consumers to buy products and publish reviews. Mining perceptual maps and radar charts from such a large number of consumer reviews may unveil majority preferences where the more satisfied the consumers are with a product feature, the higher the sentiment score the product feature has. For the results mined from both datasets, we find that the sentiment scores may be different between datasets;

[11] http://www.tomshardware.com/reviews/iphone-5-benchmark-lightning,3312-3.html.

[12] http://www.cnet.com/news/iphone-5-rated-5th-in-u-s-user-satisfaction/.

[13] http://sourcedigit.com/1913-smartphone-os-global-market-share-data-2014/.

[14] http://healthandfashion.us/why-android-phones-become-so-popular/.



STORAGE OPERATING SYSTEM PROCESSOR

NETWORK APPLICATION CASING

CP VALUE SERVICE USER EXPERIENCE

DISPLAY BATTERY ACCESSORY

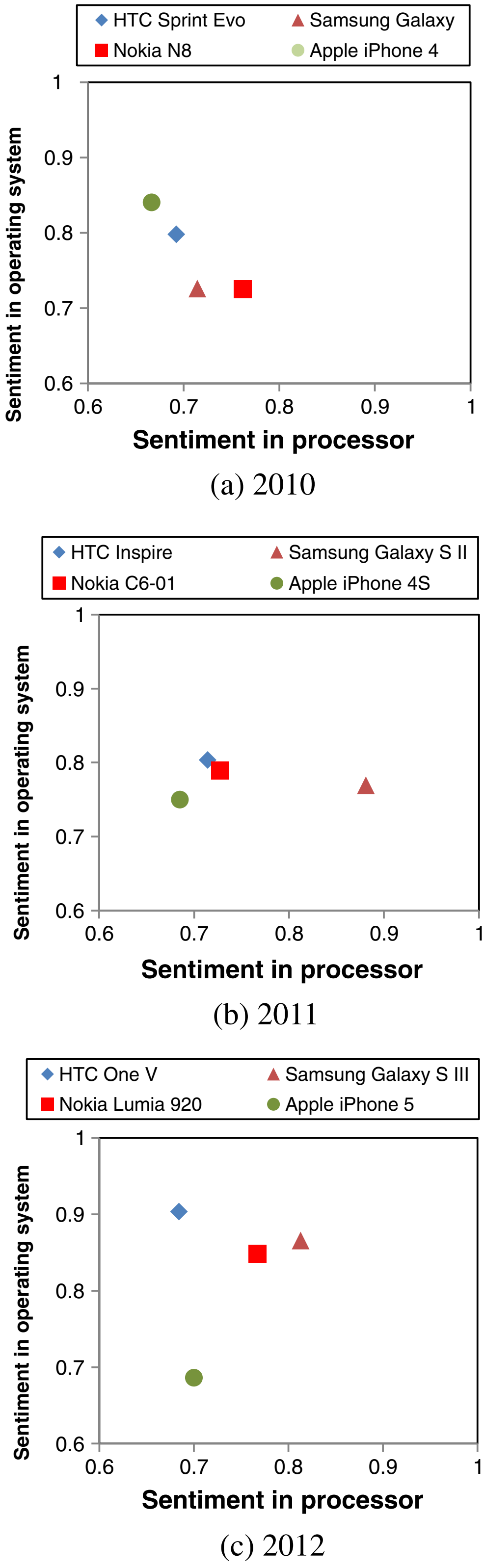
Fig. 8. Product features in each feature set.

however, most findings are similar to each other. This demonstrates the robustness of the insights and also indicates that MPM is reliable to learn majority preferences of consumers that are helpful for company's decision making.

In this study, we focus on consumer reviews of smartphones. However, MPM can be easily applied to analyze consumer reviews of other products in product positioning and marketing. What MPM requires is a few number of predefined product features as prior knowledge which can be accessible with a reasonable amount of efforts. For example, it may not be difficult for companies or experienced users to name a set of predefined product features of interests. In addition, there are more and more consumer review websites providing a list of predefined product features where these predefined features might be usually the concerns of consumers. Once the predefined product features are obtained, MPM can be easily applied to analyze consumer reviews of the products to generate business insights. Moreover, MPM can be used to analyze the consumer reviews on an individual social network or the aggregated one from a variety of social networks such as forums, twitters or blogs. By aggregating the consumer reviews from different social networks, we may find the majority preferences in these social networks. Furthermore, we may check the reliability of findings by comparing various individual datasets. Therefore, MPM can be easily generalized to analyze the consumer reviews of other products.

6. Conclusions and future work

In this paper, we have proposed a framework to mine perceptual maps from consumer reviews. The framework contains four phases, namely, extracting candidate product features with POS tags, building the virtual document for each candidate product feature, using WLDA to cluster similar product features to a feature set, and building perceptual maps and radar charts. We devise a pruning strategy to remove the irrelevant surrounding words while constructing virtual documents, where the virtual documents can capture the distributional and lexical similarities of product features. Also, we propose a weighted scheme and hard constraints in WLDA, where the product features in a constraint are guaranteed to be grouped into a feature set and the weighted scheme helps cluster together the product features with similar distributions of surrounding words and lexical meanings into a feature set. Thus, WLDA outperforms the previous methods. Moreover, by automatically building perceptual maps and radar charts from consumer

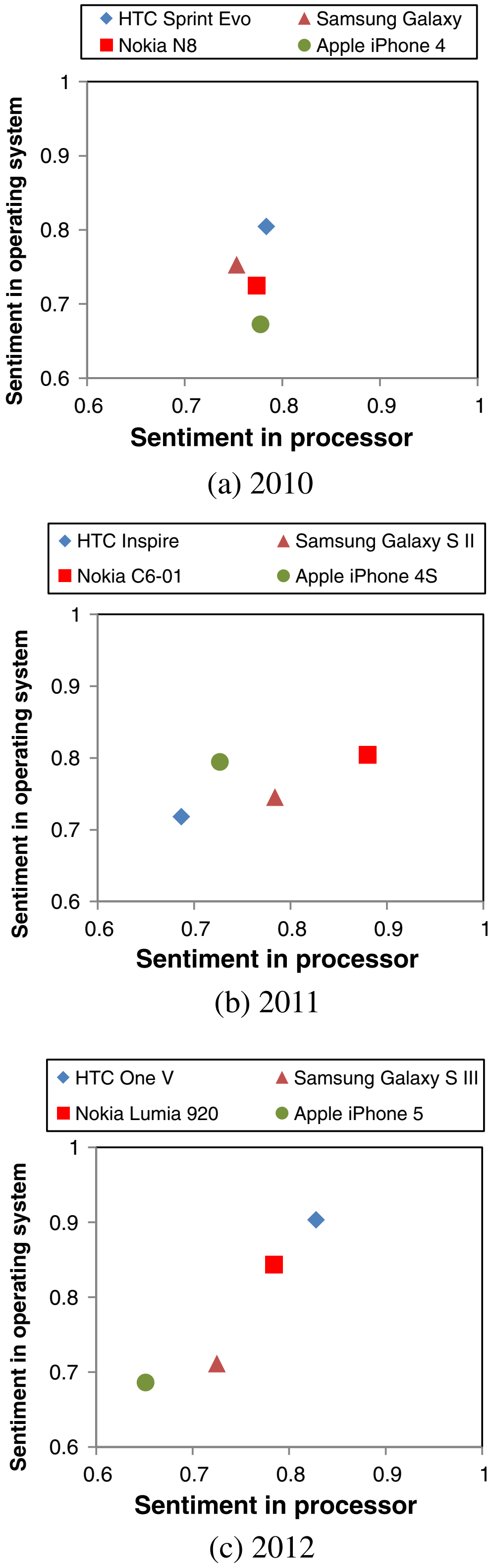


(a) 2010

(b) 2011

(c) 2012

Fig. 9. OS vs. processor by brand.



(a) 2010

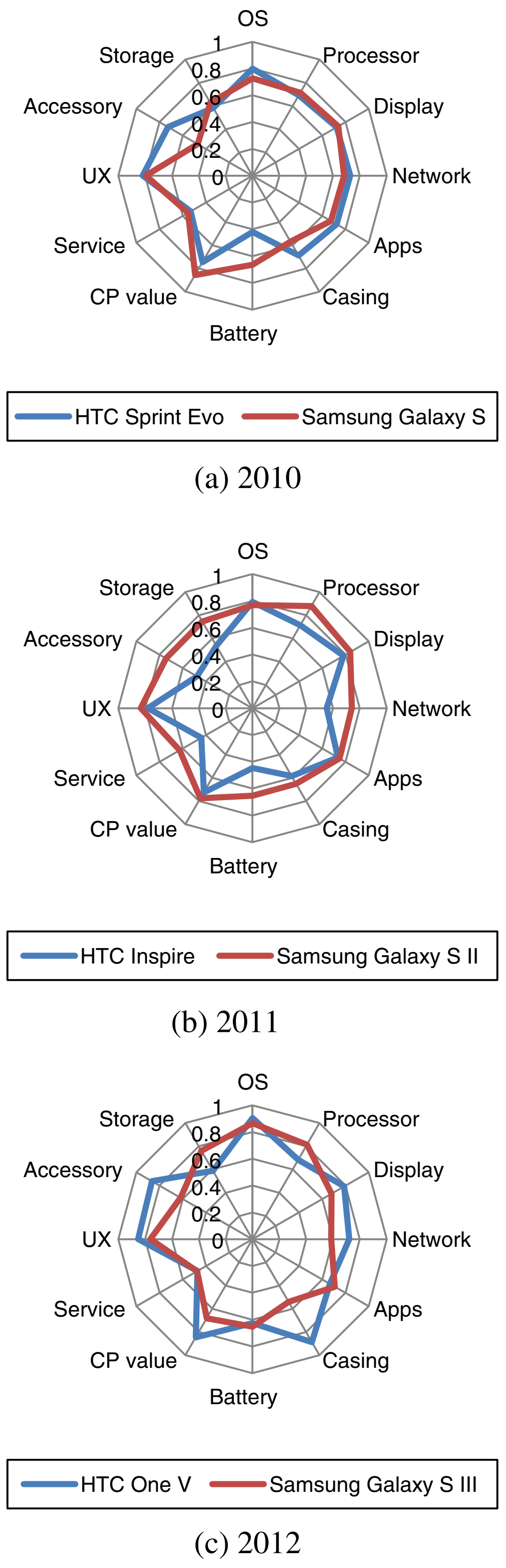
(b) 2011

(c) 2012

Fig. 10. OS vs. processor of CBLDA.

reviews, it is helpful for a company to position its products and formulate competitive strategies.

Although the proposed method uses consumer reviews of smartphones to build perceptual maps and radar charts in the experiment, it can be easily applied and extended in several aspects. First, it can be applied to analyze consumer reviews of different categories of competitive products or brands. For example, we may find ideal restaurants from analyzing the reviews of food restaurants with respect to relish and service. Second, repositioning is often another important promotional goal for companies. By comparing the existing and the



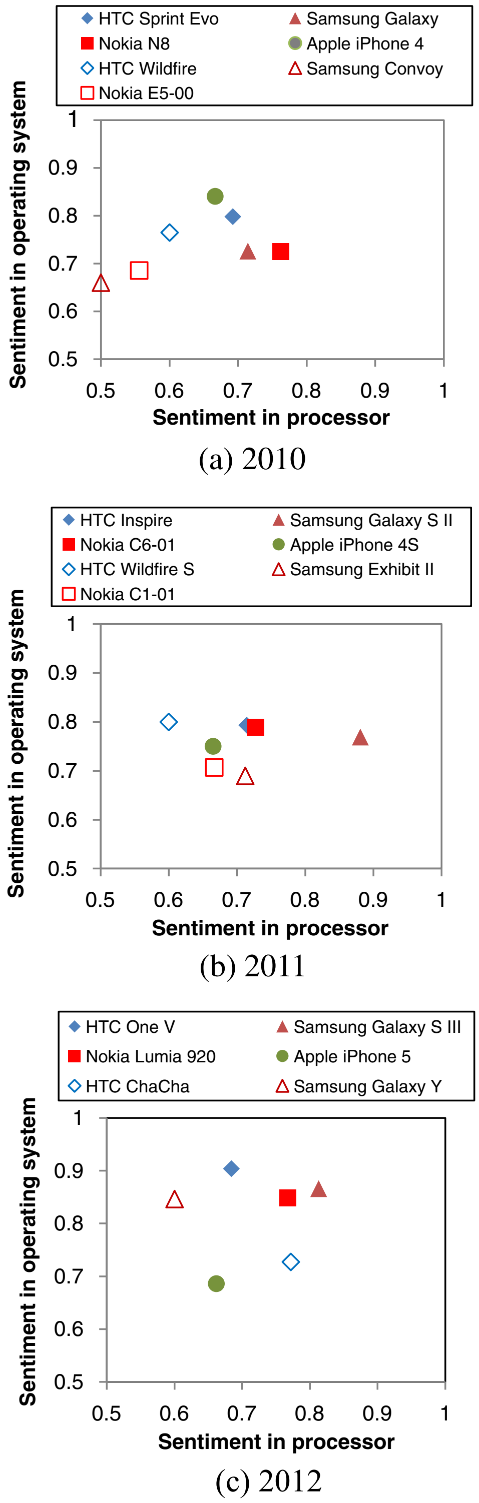
(a) 2010

(b) 2011

(c) 2012

Fig. 11. Radar charts of Samsung and HTC.

previous positioning on the same map, it may be convenient to investigate if the proposed business strategy is successful. Third, for smaller brands or companies, aligning their products as closely as possible to the market leader is a common marketing strategy so that consumers may not tell the difference of products. Thus, perceptual maps and radar charts may help smaller companies adopt a me-too positioning.



(a) 2010

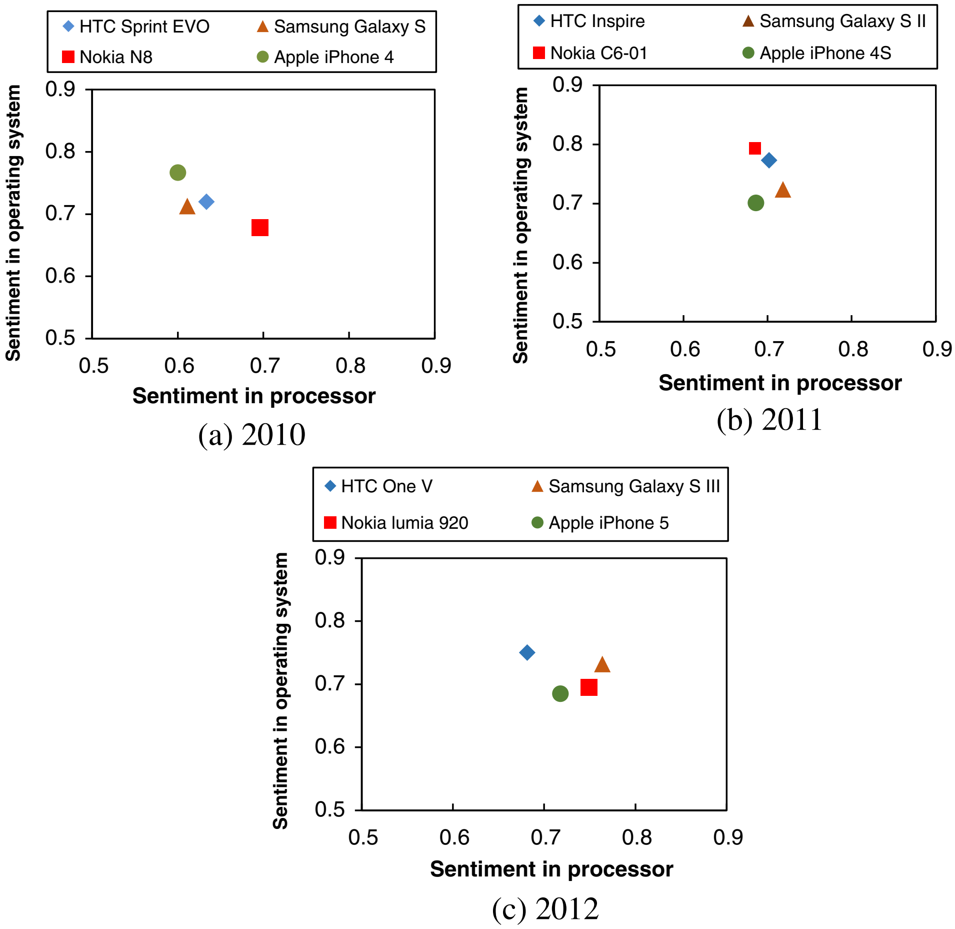
(b) 2011

(c) 2012

Fig. 12. OS vs. processor by price.

Since perceptual maps and radar charts can automatically be built from consumer reviews in real time, this information provides companies prompt business insights for decision making. Therefore, building perceptual maps and radar charts from consumer reviews can help companies gain knowledge of their products and competitors' products, which may serve good indicators to develop new products or services.

In addition to the wide applications of perceptual maps and radar charts, grouping similar product features is also beneficial. For example, we may extract all the review sentences associated with the product



(a) 2010

(b) 2011

(c) 2012

Fig. 13. OS vs. processor in PhoneArena dataset.

feature and further apply text mining methods to analyze the topics of what customers complain or praise about the product features. Moreover, grouping similar product features may allow potential buyers to search and compare a product to another more easily since different users may use different terms to search the same feature.

The proposed method can be extended in several directions in the near future. First, the proposed method can be applied to analyzing the other types of products, such as consumable products, such as food or electronic products. Next, although we use a sentiment dictionary to detect consumer sentiments and most sentiments can be accurately detected, it still has a room to improve the detection accuracy by using some natural language processing techniques. Finally, we only use the ratio of positive sentiment count to total sentiment count to calculate sentiment scores. It may be worth developing another method to calculate the score of each feature set.

Acknowledgments

The authors are grateful to the anonymous referees for their helpful comments and suggestions. This research was supported in part by the National Science Council of Republic of China under Grant No. MOST 103-2410-H-002-109-MY3.

References

[1] A. Bhatnagart, S. Ghose, A latent class segmentation analysis of e-shoppers, J. Bus. Res. 57 (7) (2004) 758–767.
[2] X. Bai, Predicting consumer sentiments from online text, Decis. Support. Syst. 50 (4) (2011) 732–742.
[3] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[4] S. Bose, N. Gupta, Customer perception of services based on the SERVQUAL dimensions: a study of Indian commercial banks, J. Serv. Mark. Q. 34 (1) (2013) 49–66.
[5] D. Buscaldi, J. Le Roux, J.J.G. Flores, A. Popescu, LIPN-CORE: semantic text similarity using n-grams, WordNet, syntactic analysis, ESA and information retrieval based features, Proceedings of the Second Joint Conference on Lexical and Computational Semantics 2013, pp. 162–168.
[6] J. Carrillo de Albornoz, L. Plaza, P. Gerv'as, A hybrid approach to emotional sentence polarity and intensity classification, Proceedings of the 14th Conference on Natural Language Learning 2010, pp. 153–161.
[7] J. Carrillo de Albornoz, L. Plaza, P. Gerv'as, A. D'iaz, A joint model of feature mining and sentiment analysis for product review rating, Proceedings of European Conference on Information Retrieval 2011, pp. 55–66.
[8] Y. Chang, C. Chang, K. Chen, C. Lei, Radar chart: scanning for satisfactory QoE in QoS dimensions, J. IEEE Netw. 26 (4) (2012) 25–31.
[9] C.M.K. Cheung, B.S. Xio, I.L.B. Liu, Do actions speak louder than voice? The Signaling Role of Social Information Cues in Influencing Consumer Purchase DecisionsDecision Support Systems, vol. 652014 50–58.
[10] B. Furlan, V. Batanović, B. Nikolić, Semantic similarity of short texts in languages with a deficient natural language processing support, Decis. Support. Syst. 55 (3) (2013) 710–719.
[11] P. Green, Marketing applications of MDS: assessment and outlook, J. Mark. 39 (1) (1975) 24–31.
[12] H. Guo, H. Zhu, Z. Guo, X.-X. Zhang, Z. Su, Product feature categorization with multilevel latent semantic association, Proceedings of the 18th ACM Conference on Information and Knowledge Management 2009, pp. 1087–1096.
[13] W.M. Han, S.J. Huang, An empirical analysis of risk components and performance on software projects, J. Syst. Softw. 80 (1) (2007) 42–50.
[14] N. Hu, N.S. Koh, S.K. Reddy, Ratings lead you to the product, reviews help you clinch it? Decision Support Systems, vol. 57 2014, pp. 42–53.
[15] M. Hu, B. Liu, Mining and Summarizing Customer Reviews, 2004, pp. 168–177.
[16] T. Lappas, G. Valkanas, D. Gunopulos, Efficient and domain-invariant competitor mining, Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2011, pp. 408–416.
[17] B. Liu, M. Hu, J. Cheng, Opinion observer: analyzing and comparing opinions on the web, Proceedings of the 14th International Conference on, World Wide Web 2005, pp. 342–351.

[18] Y. Matsuo, T. Sakaki, K. Uchiyama, M. Ishizuka, Graph-based word clustering using a web search engine, Proceedings of the Conference on Empirical Methods in Natural Language 2006, pp. 542–550.
[19] O. Netzer, R. Feldman, J. Goldenberg, M. Fresko, Mine your own business: market structure surveillance through text mining, J. Mar. Sci. 31 (3) (2012) 521–543.
[20] T. Pedersen, Information content measures of semantic similarity, Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2010, pp. 329–332.
[21] A. Popescu, O. Etzioni, Extracting product features and opinions from reviews, Proceedings of the Conference on Empirical Methods in Natural Language Processing 2005, pp. 339–346.
[22] R. Schmalensee, Perceptual maps and the optimal location of new products: an integrative essay, Int. J. Res. Mark. 5 (4) (1988) 225–249.
[23] I. Titov, R. McDonald, A joint model of text and aspect ratings for sentiment summarization, Proceedings of the 46th Annual Meeting of the Association of Computational Linguistic 2008, pp. 308–316.
[24] J. Trout, Positioning is a game people play in today's me-too market place, Ind. Mark. 54 (6) (1969) 51–55.
[25] W. Vanlaar, H. Simpson, R. Robertson, A perceptual map for understanding concern about unsafe, Accid. Anal. Prev. 40 (5) (2008) 1667–1673.
[26] H. Wang, Y. Lu, C. Zha, Latent aspect rating analysis on review text data: a rating regression approach, Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2010, pp. 783–792.
[27] G. Wang, J. Sun, J. Ma, K. Xu, J. Gu, Sentiment classification: the contribution of ensemble learning, Decis. Support. Syst. 57 (2014) 77–93.
[28] K. Xu, S.S. Liao, J. Li, Y. Song, Mining comparative opinions from customer reviews for competitive intelligence, Decis. Support. Syst. 50 (4) (2011) 743–754.
[29] Z. Zhai, B. Liu, H. Xu, P. Jia, Clustering product features for opinion mining, Proceedings of the 4th ACM International Conference on Web Search and Data Mining 2011, pp. 347–354.
[30] Z. Zhai, B. Liu, H. Xu, P. Jia, Constrained LDA for grouping product features in opinion mining, Proceedings of the 15th Pacific–Asia Conference on Knowledge Discovery and Data Mining 2011, pp. 448–459.

Anthony J.T. Lee received a BS degree in Information Engineering and Computer Science from National Taiwan University, Taiwan, and a Ph.D. degree in Computer Science from University of Illinois at Urbana-Champaign, respectively. He joined the Department of Information Management, College of Management, National Taiwan University and he is now a professor. His papers have appeared in *Pattern Recognition*, *Pattern Recognition Letters*, *Information Systems*, *Journal of Systems and Software*, *Information Sciences*, *Data and Knowledge Engineering*, *Expert Systems with Applications*, *Journal of Information Management*, *ACM Transactions on Management Information Systems*, *Decision Support Systems*, etc. His research interests include data mining, knowledge management, decision support systems, business intelligence, information economics, and business modeling. He is a member of ACM, IEEE, and AIS.

Fu-Chen Yang received an M.S. degree from National Chi-Nan University, Taiwan, in 2009. He is now a Ph.D. candidate in Department of Information Management, National Taiwan University, Taiwan. His research interests include data mining, text mining and social media analysis.

Chao-Hung Chen received BBA and MBA degrees in Information Management from National Taiwan University, Taiwan, in 2011 and 2013, respectively. He is working on his MS degree in Language Technologies Institute, School of Computer Science, Carnegie Mellon University. His research interests include data mining and social network analysis.

Chun-Sheng Wang received MBA and Ph.D. degrees in Information Management from National Taiwan University, Taiwan, in 1996 and 2007, respectively. He is an assistant professor in Department of Information Management, Jinwen University of Science and Technology, Taiwan. His research interests include data mining, text mining and social media analysis.

Chih-Yuan Sun received a BBA degree in Information Management from National Taiwan University, Taiwan, in 2014. She is working on her MBA degree in Department of Information Management, National Taiwan University. Her research interests include data mining and social network analysis.