A Systematic Methodology Based on Word Embedding for Identifying the Relation Between Online Customer Reviews and Sales Rank

Dedy Suryadi
Enterprise Systems Optimization Laboratory,
Department of Industrial and Enterprise
Systems Engineering,
University of Illinois at Urbana-Champaign,
Urbana, IL 61801;
Industrial Engineering Department,
Parahyangan Catholic University,
Bandung 40141, Indonesia,
e-mails: suryadi2@illinois.edu;
dedy@unpar.ac.id

Harrison Kim[1]
Enterprise Systems Optimization Laboratory,
Department of Industrial and Enterprise
Systems Engineering,
University of Illinois at Urbana-Champaign,
Urbana, IL 61801
e-mail: hmkim@illinois.edu

In the buying decision process, online reviews become an important source of information. They become the basis of evaluating alternatives before making purchase decision. This paper proposes a methodology to reveal one of the hidden alternative evaluation processes by identifying the relation between the observable online customer reviews and sales rank. This methodology applies a combined approach of word embedding (word2vec) and X-means clustering, which produces product-feature words. It is followed by identifying sentiment words and their intensity, determining connection of words from dependency tree, and finally relating variables from the reviews to the sales rank of a product by a regression model. The methodology is applied to two data sets of wearable technology and laptop products. As implied by the high predicted R-squared values, the models are generalizable into new data sets. Among the interesting findings are the statements of problems or issues of a product are related to better sales rank, and many product features that are mentioned in the review title are significantly related to sales rank. For product designers, the significant variables in the regression models suggest the possible product features to be improved. [DOI: 10.1115/1.4040913]

1 Introduction

Products are designed and manufactured to be successful in the market, i.e., customers are willing to buy the products. However, the buying decision process is complicated to observe and model explicitly. The underlying psychological processes, such as motivation, perception, learning, and memory [1], which affect a buying decision, differ by individuals and situations. To describe a general buying decision process, a five-stage model has been proposed [1]. The model is represented in a diagram shown in Fig. 1.

Based on the five-stage model, in order to trigger a purchase decision, it is essential for product designers to identify customer needs, learn the weights that customers assign to product features, and collect the feedback from customers who have had experience with the product or a similar one. For those purposes, product designers can conduct interviews, surveys, focus group discussions, etc. These methods, however, can be time-consuming, labor-intensive, and expensive [2].

As an alternative to the aforementioned conventional methods, analyzing publicly available online customer reviews is a resource-efficient method to learn customer needs and preference. Online reviews have grown to become an important source for customers to do information searches [3]. As reported in Ref. [4], 68% of online customers check at least four reviews and almost 25% of them check at least eight reviews before buying. Although there has been a stream of research dedicated to verify the authenticity of product reviews, as initialized by Jindal and Liu [5], this paper limits the scope of the research by assuming that the reviews are written voluntarily, and thus can be considered authentic [2].

In the framework of the five-stage model and an e-commerce setting, online review is one of the inputs for the evaluation of alternatives stage. The processes in the evaluation stage are hidden, but the input and the resulting purchase decision are both observable. The proposed methodology in this paper aims to systematically reveal one of the processes at the evaluation stage, i.e., assigning weights to product features. By discovering product features that are significantly related to product sales, it may be implied that those features are the ones weighted as more valuable by customers. Thus, this information provides an objective data-driven suggestion for designers about possible features to improve.

In revealing the hidden process of assigning weights to product features at the evaluation stage, the challenges are:

(1) Customers may discuss product features that are not mentioned in the product description on a product's webpage. Therefore, product descriptions are not adequate to capture the product features discussed in the reviews.
(2) Customers may discuss the same product feature using different words, e.g., "drive," "storage," and "SSD."
(3) Customers may express their opinions with their own style in free-format reviews. Thus, free-format reviews are more difficult to analyze, compared with reviews that have been distinctly divided into Pros and Cons section, such as in Ref. [6].

Regarding the challenges above, the proposed methodology needs to solve four tasks, i.e.,

(1) The methodology should obtain product-feature words (*task 1*), i.e., words that represent product features, discussed in the customer reviews.
(2) The methodology should group the same product-feature words that refer to the same product feature (*task 2*).

[1]Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received November 20, 2017; final manuscript received July 9, 2018; published online September 18, 2018. Assoc. Editor: Scott Ferguson.

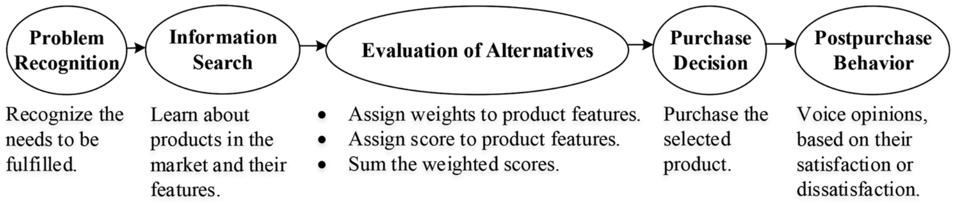

Fig. 1 Five-stage model of buying decision process

(3) The methodology should obtain words that describe sentiment, as well as the intensity of the sentiment (*task 3*).
(4) The methodology should connect each sentiment word in a sentence to the corresponding product-feature word in the sentence (*task 4*).

There has been research done on the similar topics, as discussed in Sec. 2.1. However, there are main differences between this paper and the previous works, i.e.: (1) the application of word-embedding followed by X-means clustering to obtain product-feature words, (2) the analysis of free-format review data that are not divided into Pros and Cons sections, (3) the elaboration of methods applied in each stage of the methodology, and (4) the analysis of review title as a separate variable from the review content to discover its importance compared to the content. Furthermore, subjective inputs, judgments, and decisions are kept to minimum in the proposed methodology. It does not require, for example, words as initial seeds to discover product-feature words or human judgments (e.g., crowdsourcing). Therefore, the methodology is replicable and generalizable to data sets of different products. It is an improvement to the methodology used in the initial research [7,8] that mainly relies on subjective judgments in identifying and grouping the relevant product-feature words.

This paper is organized as follows: Section 2 elaborates the previous works in the similar topic, followed by the introduction of word embedding technique. Section 3 details each stage in the proposed methodology. Section 4 describes the data used for case study, shows the results from processing the data, and finally presents the regression results. Section 5 discusses both the methodology and the results. The last section concludes the paper.

2 Literature Review

There are two main topics presented in this section, i.e., the previous works in the similar area and word embedding. Sec. 2.1 discusses numerous works done in analyzing text from online customer reviews. Sec. 2.2 introduces the word embedding technique that becomes one of the main tools in the proposed methodology.

2.1 Previous Approaches in Solving the Four Tasks. This subsection is focused on the four tasks required to interpret free-format reviews that have been presented in Sec. 1. It is worth mentioning here that there are papers that completely ignore textual contents of online customer reviews and only utilize variables such as number of reviews and star ratings [3,9]. Interestingly, one of the results in Ref. [3] suggests that customers actually read the review content. *Consequently, this paper argues that the inclusion of textual-related variables in the analysis is necessary.*

Interpreting free-format reviews starts with obtaining product-feature words. In some cases, the product-feature words have been known or predetermined, such as in Ref. [10]. When they are not known, as the case in this paper, various approaches have been applied, as summarized in Table 1. Many of those approaches rely upon manually annotated data, as well as subjective predetermination of linguistic patterns and product-feature words to obtain. The main disadvantage of the heavily manual approaches is that they may not be generalizable into the data from other domains.

The other approaches to solve task 1 apply association rule, tf.idf (term frequency, inverse document frequency), and latent Dirichlet allocation (LDA). Association rule is used to find frequent *itemsets*, i.e., words or phrase that occurs together frequently [11,12], which are assumed to be product-feature words. In fact, that might not be the case and the proposed pruning rules are not able to filter the irrelevant *itemsets* [13]. The same disadvantage applies to the approach that assumes words with high tf.idf to be product-feature words [14]. For the LDA-based approaches [15,16], the main disadvantage is the necessity to determine the number of topics beforehand. In the case of online reviews, the number is not known beforehand, because customers might discuss product features that are not described in the product's webpage. *Considering the disadvantages of the previous approaches, the proposed methodology aims to obtain product-feature words with as little manual involvement as possible and exploit the review data to guide the process.*

Performing task 1 often returns an unmanageable number of product-feature words. However, in fact, many of those words refer to the same product feature, e.g., "screen" and "monitor."

Table 1 Summary of previous approaches and their disadvantages to solve task 1

References	Approaches	Disadvantages
[17]	Subjective determination	Depending highly on the person who annotates the corpus
[18,19]	Supervised machine learning tools: Decision Stump, Conditional Random Fields	Requiring manually annotated or tagged training data
[11,12,20,21]	Association rule to find noun or noun phrases that frequently appear together	Resulting in nouns with high frequency, but not related to product features [13]
[13]	Association rule with additional filtering step using a set of "subjective adjectives"	Requiring manually constructed set of "subjective adjectives", which can be domain-specific for a particular type of products
[14]	High tf.idf (term frequency, inverse document frequency) rule	Resulting in nouns with high tf.idf, but not related to product features
[22,23]	POS patterns	Requiring manually determined POS patterns to mine
[24]	Hidden Markov Model, based on tags of product-feature and sentiment words	Requiring manual tagging of training data
[15]	Latent Dirichlet allocation	Requiring predetermined number of topics to generate (in this approach, a topic corresponds to a product feature)
[16]	Augmented LDA to learn both product-feature and sentiment words	Requiring predetermined number of topics to generate

Table 2 Summary of previous approaches and their disadvantages to solve task 2

References	Approaches	Disadvantages
[17]	Subjective grouping	Depending highly on the person who groups the words
[23]	Product ontology	Requiring manually constructed ontology
[6]	Multilevel LDA	Requiring predetermined number of topics to generate
[21]	WordNet-based similarity	Requiring word sense disambiguation to use WordNet in order to determine the correct similarity between a pair of words
[20]	WordNet-based similarity and agglomerative clustering	Requiring word sense disambiguation to use WordNet and the details for clustering are not provided;
[25]	Lexical similarity and Expectation Maximization	Assuming good quality product-feature words have been obtained and the number of groups is known beforehand

Table 3 Summary of previous approaches and their disadvantages to solve task 3

References	Approaches	Disadvantages
[18,28]	Manual annotation of the sentiment words and polarity	Depending highly on the person who annotates the corpus
[20]	Amazon mechanical turk	Depending highly on the people who join the crowdsourcing
[11,12]	The adjective closest to a product-feature word is considered as a sentiment word	Assuming an adjective always modifies the closest noun; not using sentiment intensity quantification
[17]	The most frequent adjectives are collected, and considered as sentiment words	Requiring a subjective threshold for the frequency of adjectives; not using sentiment intensity quantification
[10]	Pattern-based search	Requiring initial patterns, e.g., "the (feature) is (sentiment)"
[14]	Senti-WordNet (the complement of WordNet, with added sentiment polarity)	Requiring word sense disambiguation to use WordNet in order to determine the correct sense for the word on hand
[23]	Dependency tree	Not using sentiment intensity quantification

Therefore, for the purpose of interpreting the reviews as accurately as possible, it is essential to solve task 2. Table 2 summarizes the previous approaches to solve task 2.

It can be seen from Table 2 that many approaches require subjective decisions, such as determining the number of product-feature groups. In reality, the number of product features that are discussed in the product reviews is initially unknown. Therefore, a clustering tool such as K-means clustering is not suitable, because the number of clusters K needs to be determined. On the other hand, X-means clustering does not require the number of clusters as an input. Iteratively, X-means clustering splits a cluster temporarily into two and computes the Bayesian information criterion (BIC) measure in Eq. (1). A cluster is permanently split only if there is an improvement from splitting, i.e., the BIC value increases. The iteration stops when there is not a split of clusters that increases the BIC value. Therefore, the clusters can be finally obtained without predetermining the number of clusters.

$$BIC(M_j) = \hat{l}_j(D_{BIC}) - \frac{p_j}{2}\log(R) \tag{1}$$

One of the objective approaches in Table 2 is using WordNet-based similarity. However, a similarity-based approach in WordNet requires word sense disambiguation [26] technique to obtain the correct similarity between a pair of words. For example, the similarity between the words "battery" and "computer" in WordNet depends on the sense of both words. If "battery" is defined in the sense of "a device that produces electricity" and "computer" is "a machine for performing calculations automatically," then the similarity is significantly higher than if "battery" is defined in the sense of "an assault in which the assailant makes physical contact." *Therefore, in order to overcome those aforementioned disadvantages, this paper combines a word embedding and X-means clustering approaches in order to solve tasks1 and 2 automatically and objectively, i.e., without manually annotating training data, predetermining linguistic patterns, or predetermining the number of product-feature words.*

While tasks 1 and 2 deal with product-feature words, task 3 deals with sentiment words. Table 3 summarizes the previous approaches to solve task 3. The disadvantages of those approaches are the reliance on manually annotated data, subjective inputs (e.g., initial seeds of patterns), and word sense disambiguation. *In order to overcome those disadvantages, this paper simply identifies the adjectives as sentiment words. The identification of adjectives is objectively obtained from a part-of-speech (POS) tagger.* Furthermore, in order to capture a customer's sentiment more accurately, it is important to quantify the sentiment intensity. For example, a comment of "*great* battery" is more intense than "*good* battery." *In this paper, the sentiment intensity quantification is obtained from SenticNet4 dictionary, which captures the denotative and connotative information associated with objects, people, actions, and events* [27].

Finally, after product-feature words are grouped (task 2) and sentiment words are identified (task 3), the correct connection between those words needs to be identified (task 4). In the previous works, other than manual mapping of the connection between sentiment word and its corresponding product-feature word [28], either distance or dependency is used to infer the connection. In the distance-based approach, a sentiment word is simply connected to the closest product-feature word [11,12]. In the dependency-based approach, several rules are applied to a dependency tree in order to obtain connected product-feature and sentiment words [17,18,20,23]. Regardless of the distance between words in a sentence, a dependency tree is capable to show the words that are related, as discussed later in Sec. 4 and presented in Fig. 2. *Considering its advantage compared to the distance-based approach, this paper uses a dependency tree to infer the connections between product-feature and sentiment words.*

2.2 Word Embedding. Word embedding is a distributed representation for words in a vector space [29]. It is based on the idea that similar words have similar distribution of words that are likely to appear along with them. Therefore, the vectors representing similar words should be similar as well. Zhang et al. [30] has applied word2vec, a word embedding tool, to retrieve synonyms of a given set of product-feature words. *In this paper, however, the product-feature words are initially unknown. The word*

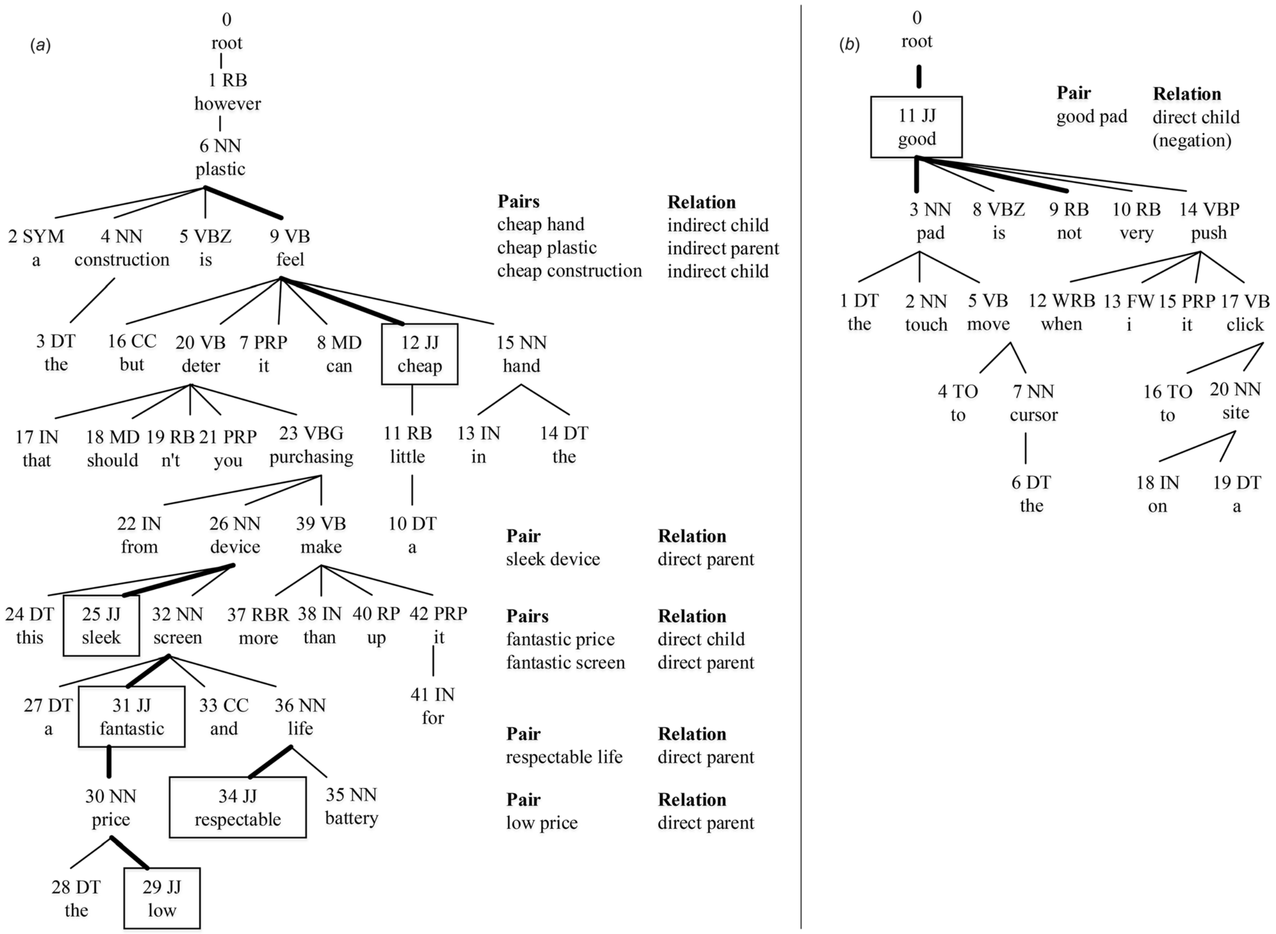

Fig. 2 Relations between adjectives and nouns in: (*a*) a sentence without negation and (*b*) a sentence with negation

embedding tool is then applied to discovering those words from customer reviews.

Skip-gram model is introduced by Mikolov et al. [29] to learn good word embedding. For training the model, the objective is to maximize the probability of context words given a target word. Context words are commonly defined as the words surrounding a target word within a window of words. Context words can be defined differently, such as considering the dependency relations [31], but this paper uses the window-based definition.

The training is done through a neural network. As illustrated in Fig. 3, the network consists of three layers, i.e., input, hidden, and output layers. In the input layer, each element of a V -dimensional vector represents a word in the vocabulary, where V is the vocabulary size. When a particular target word is used to train the network, a particular position in the vector that corresponds to the word is set as 1, while the value of other elements remains 0. This vector is transformed by an input matrix W into an N-dimensional vector in hidden layer. Each row in W represents the embedding of a word. From hidden layer, the vector is further transformed by an output matrix W' into the output layer. Suppose the window for context words is C, then there are C output vectors in the output layer.

Suppose the target word I is given and the jth element of the ith output vector is denoted as $u_{i,j}$, then $y_{i,j}$ is the probability of word j being a context word is calculated by Eq. (2). The objective of the model is to maximize Eq. (2) with respect to a particular context word $j*$, which is the actual context word of target word I. By going through the sentences in the corpus, the network learns and updates the matrices W and W', such that words that have similar context words are expected to have similar vector representations in W. The detailed derivation of the gradient descent formulas to update the matrices is presented in Ref. [32].

$$P\left(w_j|w_I\right) = y_{i,j} = \frac{\exp(u_{i,j})}{\sum_{j'=1}^{V} \exp\left(u_{i,j'}\right)} \tag{2}$$

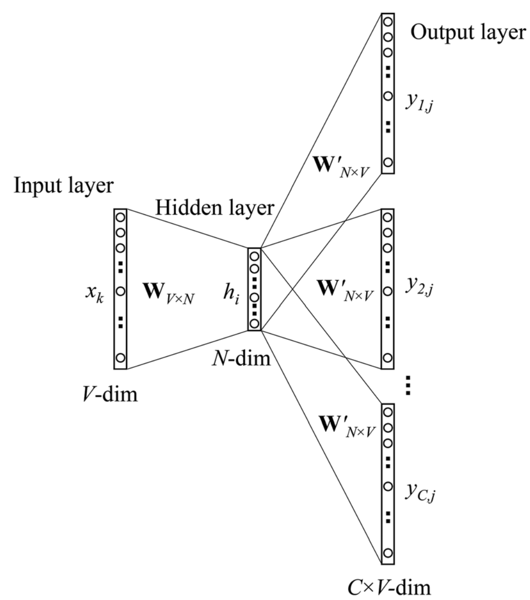

Fig. 3 Skip-gram model (Source: [32])

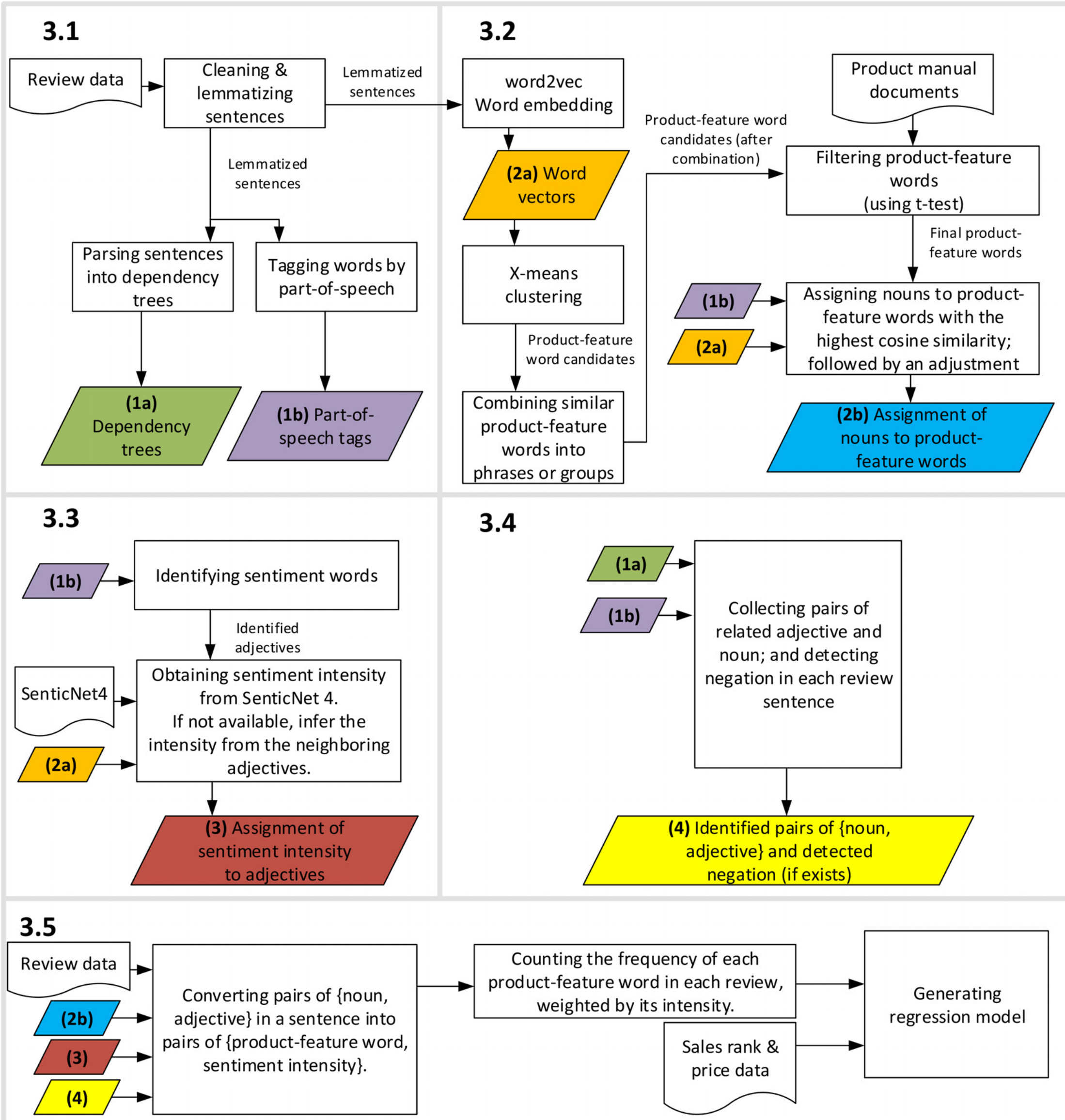

Fig. 4 The flowchart of the proposed methodology

3 Methodology

The proposed methodology is presented as a flowchart in Fig. 4. The numbers on the flowchart correspond to the corresponding subsection numbers in this paper.

3.1 Data Preprocessing. The first step in preprocessing the review data is removing nonalphanumeric characters, such as #, $, and %. These characters are considered not helpful to reveal either product-feature or sentiment words from a sentence. Afterward, a lemmatizer, which is obtained from natural language toolkit package in PYTHON, is applied to replace various word forms into their basic forms, e.g., replacing a word in plural form "years" into "year." The replacement is required to avoid having the same word in different forms embedded into different vectors.

Furthermore, each sentence in the customer reviews is parsed into a dependency tree. A dependency tree describes the structure of a sentence by relating words in terms of binary semantic or syntactic relations [26]. Therefore, each link in the tree explains the relation between two words. In this paper, the dependency tree is obtained using PyStanfordDependencies package in PYTHON [33]. The trees become the inputs for Sec. 3.4. Other than the dependency relation, the parser also provides the part of speech for each word. The relevant POS tags for the purpose of this paper are nouns (NN, NNS), proper nouns (NNP, NNPS), and adjective (JJ), which become the inputs for Secs. 3.2–3.4.

3.2 Product-Feature Words Identification. This subsection is divided into two parts, i.e., identifying the initial product-feature words and filtering out the irrelevant words to obtain final product-feature words.

3.2.1 Initial Product-Feature Words Identification. In this stage, a word embedding tool word2vec is used to obtain product-feature words. The input for word2vec is the lemmatized sentences from customer reviews. The parameters to be determined for word2vec are the dimensions of the embedding vector, the window size for the context words, the cutoff frequency of words, the usage of either hierarchical softmax or negative sampling [32], and the initial random seed—in order to create a fully replicable result. The word2vec used in this paper is obtained from gensim [34] package in PYTHON. The output from word2vec is the representation of words in vectors in real numbers.

The vectors output by word2vec are subsequently clustered with X-means clustering technique [35]. Since product-feature words are assumed to be nouns, as in the previous literatures

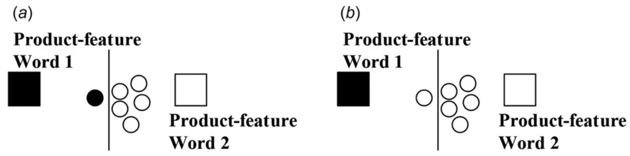

Fig. 5 Word assignment into clusters: (*a*) before adjustment and (*b*) after adjustment

[6,11,14,15,17,20,23], it is worth noting that there are typically thousands of nouns discovered from the reviews. All of the nouns, however, are not equally important. For example, in the customer reviews for laptops, the word "laptop" is arguably more important than "dog," although both words may appear in the reviews. In order to reflect the difference in importance, each word is weighted by its tf.idf (term frequency, inverse document frequency). The formula in Eq. (3) is modified from Ref. [36], such that it captures the importance of a word with respect to all documents. The computed tf.idf is incorporated into X-means clustering as the weight for each word.

$$ti_i = (tf_i)idf_i = (tf_i)\log\frac{|D|}{|d : w_i \in d|} \quad (3)$$

Based on the X-means clustering result, the word whose vector is the closest to each cluster center is determined as the product-feature word. In order to avoid redundancy caused by highly similar product-feature words, those words are either combined into a phrase or grouped together. For example, the words "heart" and "rate" are combined into a phrase "heartrate"; because the cosine similarity between "heart" and "heartrate," as well as "rate" and "heartrate," is higher than "heart" and "rate." Therefore, the phrase "heartrate" is considered as the product-feature word to represent both "heart" and "rate." Furthermore, for the remaining product-feature words, if the cosine similarity between two words is higher than a similarity threshold, then they are still grouped together but not as a phrase, e.g., "web" and "internet" become "web-internet."

3.2.2 Product-Feature Words Filtering. At this stage, a set of product-feature words have been obtained. However, the set may contain a word that has a high tf.idf value, but not related to product features, e.g., "son." An objective method to filter out such words is proposed. The input for this method is a set of manual documents of the products. In order to avoid bias of overweighting words, which are specific to a particular brand of product, it is suggested to select one manual document for one brand of product. Based on each manual document, the proportion of a product-feature word w is computed, i.e., the frequency of the word w divided by the total number of words in the document. Afterward, a one-sample t-test is performed with the null hypothesis stating that the average proportion equals zero and the alternate hypothesis stating that the average proportion is greater than zero. If the hypothesis is rejected, then the word w has an average proportion that is significantly not equal to zero; i.e., the word is common enough to appear in manual documents and thus it is likely to be a representative product-feature word. Otherwise, the word w is eliminated from the product-feature word set.

After the filtering, all other words are assigned to the final set of product-feature words based on the highest cosine similarity. As the final refinement, the assignment of a word is adjusted based on the other words similar to it. The underlying assumption for this adjustment is that similar words tend to belong to the same product feature. The process of adjusting the cluster for a word is illustrated with a simplistic two-dimensional plot in Fig. 5. It is illustrated that a word (represented as a black dot in Fig. 5(*a*)) is initially assigned to product-feature word 1, because it has a higher similarity with product-feature word 1 than product-feature word 2. However, the other words similar to it (represented as white dots in Fig. 5) are assigned to product-feature word 2. Therefore, the adjustment is made by re-assigning the word from product-feature word 1 to product-feature word 2, as shown in Fig. 5(*b*).

3.3 Sentiment Intensity Quantification. In this stage, the purpose is to identify sentiment words and quantify their intensity. Based on the part-of-speech tagging result in the Data Preprocessing stage, the adjectives are identified as sentiment words. The sentiment intensity of an adjective is obtained from SenticNet4. Originally, SenticNet is a sentiment dictionary that is developed based on combining ConceptNet and WordNet-Affect [37]. In SenticNet4 [27], a sentiment intensity score is assigned to each concept, such as 0.664 for "good," 0.179 for "okay," −0.530 for "faulty," and −0.900 for "terrible."

As thorough as it is, there are words that are not included in the SenticNet4 dictionary. The sentiment intensity for each of these words is then obtained by weighted averaging the intensity of the adjectives similar to it, as shown in Eq. (4). Similar adjectives are identified from the word vectors that are obtained in Sec. 3.2. The assumption is that similar words, including adjectives, should be embedded close to one another and thus the intensity may be inferred by the surrounding words. This inference makes it possible to obtain sentiment intensity for a new or informal adjective

$$\mathrm{int}(w) = \sum_{s \in S_w} sim(w, s)(\mathrm{int}(s)) \quad (4)$$

3.4 Dependency Tree Interpretation. At this stage, the correct connection between a product-feature word (a noun) and a sentiment word (an adjective) in a sentence needs to be discovered using a dependency tree. The connection is discovered by interpreting the dependency tree as follows: (1) The pairs of adjectives and nouns which are directly connected as parent and child become the output of this stage, (2) The adjective that has no nouns as either its direct parent or child performs further search toward the root of the sentence to discover indirect parents and children. The existence of negation words in a sentence is also important, because it may flip the sentiment expressed toward a product feature. If there is a negation word connected to an adjective and noun pair, then that pair is marked as having a negation.

The possible connections of adjective and nouns in a dependency tree are illustrated in Fig. 6. The figure shows the following possible relations: (*a*) a noun is the direct child of the adjective, (*b*) a noun is the direct parent of the adjective, (*c*–1) the adjective has no nouns as either the direct child or parent, (*c*-2) the adjective moves toward the root and replaces its current direct parent, hence the new shaded box with "(JJ)" label; the adjective now has a noun as its parent (indirect parent), (*c*3) the adjective moves further toward the root; the adjective now has a noun as its child (indirect child). The existence of a negation word negates the relations accordingly, as shown with the bold lines in the figure.

3.5 Regression Model Generation. At the last stage, in order to discover variables that are significantly related to sales rank, a linear regression model is used to link all the variables with sales rank. Previously, a linear regression model has been used in Refs. [3], [20], and [38] for the same purpose, assuming a linear relationship between the dependent and independent variables. The assumption is taken because determining the best regression model among a massive number of possibilities requires a massive

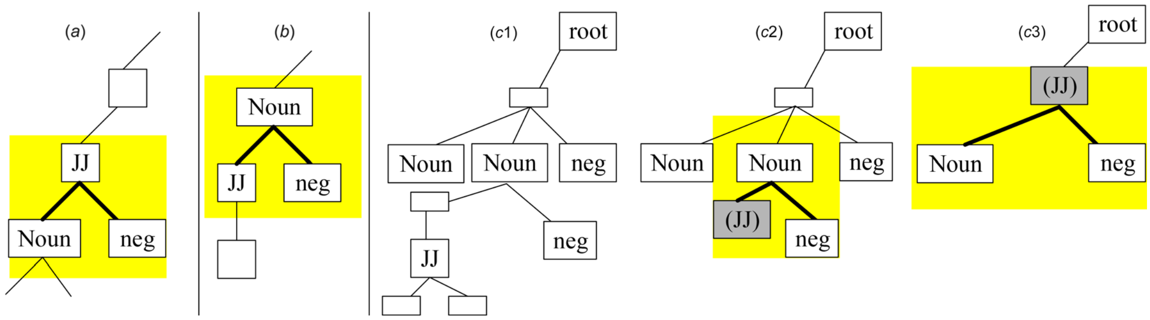

Fig. 6 Connecting adjective (JJ) to nouns in a sentence: (*a*) direct child, (*b*) direct parent, (*c*-1) no relations found, so the search continues to (*c*-2) and (*c*-3) by moving the JJ toward the root; (*c*-2) indirect parent; (*c*-3) indirect child

computational time. Furthermore, it also depends on the data sets, because different data sets may show different behavior in the relationship.

The dependent variable for the regression model is the log of the sales rank of a product at a particular time. The log of sales rank is justified because, as reported in Ref. [3], the relationship between log sales rank and log sales is close to linear. The aforementioned papers [3,20,38] use the log of sales rank as dependent variable. The independent variables are price of the product, as well as the textual and nontextual variables from the reviews. Textual variables are the count of positive or negative comments toward a particular product feature in the reviews. Nontextual variables are the average number of verified purchases, the average star ratings, the average length of reviews, the number of reviews, the percentage of reviews with a good rating (4 and 5 star ratings), and the percentage of reviews with a bad rating (1 and 2 star ratings). As in Ref. [3], the sales rank of the previous day is excluded from independent variables in the regression model. By excluding it, the model reveals more about the relations between review and sales rank. Otherwise, the explanation of variance in sales rank is highly dominated by the sales rank of the previous day. Thus, the regression model can be defined as in Eq. (5).

$$\begin{aligned}\ln(\mathrm{Rank}_{i,t}) = {} & \mathrm{Price}_{i,t} + (aVer_{i,T} + aRat_{i,T} + nRev_{i,T} + pF_{i,T} \\ & + pO_{i,T} + a\mathrm{Len}_{i,T}) + (fP_{f,i,T} + fN_{f,i,T} \\ & + tFP_{f,i,T} + tFN_{f,i,T}) + \nu_i \end{aligned} \tag{5}$$

The independent variables may correlate to one another; therefore, stepwise regression is applied in order to avoid highly correlated variables entering the model. Stepwise regression is an algorithm to select a subset of variables in a regression model. The first dependent variable selected into the subset is the one with the highest correlation with the independent variable. The next variable is added into selected set if the ratio of residual sum of squares decrease is greater than an "F-to-enter" value. In addition, any variable in the selected set can be dropped if the ratio of residual sum of squares increase is less than an "F-to-drop" value. The details of the algorithm can be found in Ref. [39].

For the performance measures of the regression model, two types of R-squared measures are used. The first measure is adjusted R-squared, which provides the percentage of variation in the data explained by the regression model. The value is adjusted with the number of independent variables in the model. The second measure is predicted R-squared. It describes how well the model predicts responses for new observations. This is calculated using the predicted residual error sum of squares statistic and total sum-of-squares (SST) in Eq. (6). It can be seen that predicted R-squared is a leave-one-out cross-validation technique. If the predicted R-squared is significantly lower than the adjusted R-squared, then the regression model overfits the training data, i.e., the model would not generalize well to a new observation or a new data set.

$$R^2(\mathrm{pred}) = \left(1 - \frac{\mathrm{PRESS}}{\mathrm{SST}}\right)100 = \left(1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_{i,-i}\right)^2}{\sum_{i}\left(y_i - \bar{y}\right)^2}\right)100 \tag{6}$$

4 Case Study

The methodology proposed in Sec. 3 is applied to two data sets. The data sets correspond to wearable technology products and laptops that have webpages in the website link.[2] They were chosen because wearable technology products were launched about a decade ago, while laptops have been in the market for a longer time and thus the features have been familiarly known by most people. Furthermore, the consideration for the chosen products is that they need to have an adequate and stable stream of reviews such that they can be related to the sales rank data. This section describes the data sets, presents examples of results from applying each stage of the methodology, and finally reports the regression results.

The data are accessed from in the website link,[2] parsed using urllib parser and organized by beautifulsoup package in PYTHON. There are 83,565 reviews for wearable technology products and 66,172 for laptops, which were written during the period of January 2015 to February 2017. An example of a review data is shown as follows:

Title: "Five Stars",
Review: "Great computer. Love it!",
ProductName: Acer Aspire E 11 ES1-111M-C40S 11.6-Inch Laptop (Diamond Black),
Direct URL: http://www.amazon.com/gp/customer-reviews/R20J53OBD5MTNO/ref=cm_cr_arp_d_rvw_ttl?ie=UTF8&ASIN=B00MNOPS1C,
Month: 02, Year: 2017, Date: 16, Verified: true, Helpful: 0, Rating: 5.0

The sales rank data record the periodic sales rank and its corresponding price. For wearable technology products, the data were collected in two periods, i.e., September 2015 to April 2016 and September 2016 to February 2017. At the beginning, there were 140 products whose data were collected. However, in order to keep the ranking consistent in the same category, only items which are ranked in the "Clips, Arms, and Wristbands" category are kept. Furthermore, the duplicated webpages of a product with

[2] Amazon.com

Table 4 Examples of vector representations for selected words, with the cosine similarity with respect to the word" display"

Word	$d=1$	$d=2$	...	$d=100$	Cosine similarity
Display	−0.419401556	0.673747182	...	−0.773826361	1
Screen	−0.205376133	0.451731592	...	0.198629543	0.65799
Storage	−0.443754196	−0.346733302	...	−1.134292126	−0.03138

Table 5 Product-feature words (wearable technology products)

Category	Words
Final product-feature words (15)	Activity, alarm, battery, button, charge, clip-strap, company-support-service, data, day, fitness-pal, heart-rate, phone-laptop-app, problem, screen, wrist-band.
Filtered Out Words (8)	Bra, money, monitoring, plastic, shade, sister, sleep, yoga.

Table 6 Product-feature words (laptops)

Category	Words
Final product-feature words (18)	Apps, battery, cable, card, drive, fan, issue, laptop, life, network, office, performance, resolution-quality, screen-display, service, supervisor, track-mouse, web-internet.
Filtered out words (10)	Asus, browsing, casing, cd, everything, Facebook, macbook, memory, son, week.

Table 7 Word assignment before and after adjustment for selected product-feature words (wearable technology products)

	Phone-laptop-app		Problem		Wristband	
Product-feature word	Before	After	Before	After	Before	After
1	device	app	issue	issue	tracker	device
2	app	phone	problem	problem	band	band
3	phone	use	review	reason	one	watch
4	tool	work	complaint	complaint	watch	wrist
5	user	apps	motivator	deal	wrist	unit

different sizes or colors are removed, because the webpages share the same reviews. Finally, 35 unique products remain. For laptop products, the data were collected in the periods of October 2015 to June 2016 and November 2016 to February 2017. The collection was started by choosing the laptops listed on the Top 100 and the ranking is recorded according to the "Traditional Laptop" category. Finally, after the removal of discontinued items, 84 products remain in this data set.

4.1 Processing Review Data. This subsection is divided into two parts, i.e., processing review data to obtain product-feature words and obtaining connections between the product-feature and sentiment words using dependency tree.

4.1.1 Obtaining Product-Feature Words. After being preprocessed, the words from the reviews become the input for word2vec. Since there has been no strict guidelines for determining the optimal parameter values in word2vec, the word2vec parameters are set based on the observations of the preliminary experiment results. For the data set of wearable technology products, the dimensions of the word embedding vector are 100, the window size is 3, the cutoff frequency is 8, hierarchical softmax is used, and the initial random seed is 0. For the data set of laptops, the same set of parameters is used, except the window size is 2. Table 4 shows examples of the vector representations of words from laptops data set. It can be seen that the representations successfully achieve a higher similarity for the pair of similar words ("display" and "screen") than the other pair ("display" and "storage").

In the stage of obtaining product-feature words, X-means clustering is performed by the pyclustering package in PYTHON[3] and it outputs cluster centers. The words closest to the centers become the initial product-feature words. The similar product-feature words are combined into phrases or groups such that 14 out of 31 words (45.16%) are combined in wearable technology products and 8 out of 32 words (25.00%) are combined in laptops. After filtering out the words that are not related to product features (e.g., "sister," "son") and specific to particular brands (e.g., "asus," "macbook"), the final results are shown in Tables 5 and 6.

Once the final product-feature words have been obtained, all nouns can be assigned to the product-feature words based on the highest cosine similarity. After improving the assignments, according to the adjustment procedure shown in Fig. 5, the group of words under the same product-feature word becomes more cohesive as presented in Tables 7 and 8. The tables display five most similar words to the corresponding product-feature words. To highlight the contribution of the adjustment to the cohesiveness, several movements are provided as examples here, i.e., "device" moves from "phone-laptop-app" to "wristband" in Table 7 and "keyboard" moves from "screen-display" to "track-mouse" in Table 8. Quantitatively, the adjustment procedure produces a higher average similarity between words within a group. The average cosine similarity between words within a

[3]https://github.com/annoviko/pyclustering

Table 8 Word assignment before and after adjustment for selected product-feature words (laptops)

Product-feature Word	Resolution-quality		Screen-display		Track-mouse	
	Before	After	Before	After	Before	After
1	Size	Size	Screen	Screen	Key	Keyboard
2	Quality	Quality	Keyboard	Color	Mouse	Key
3	Value	Speaker	Color	Display	HP	Bit
4	Speaker	Sound	Display	Pad	Touchpad	Mouse
5	Resolution	Resolution	Pad	Picture	Case	Reason

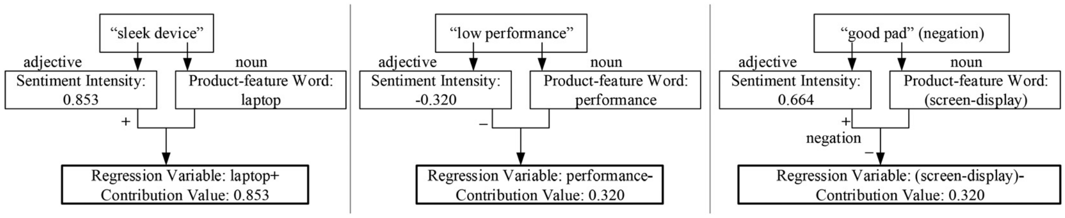

Fig. 7 Conversion into regression variables

group increases 42% for wearable technology products (from 0.1533 to 0.2176) and 38% for laptops (from 0.1210 to 0.1667).

4.1.2 Obtaining Connections Using Dependency Tree. The determination of relations between adjectives and nouns in a sentence relies on a dependency tree. As an example, the dependency tree for the sentence "*however a the construction is plastic it can feel a little cheap in the hand but that shouldn't deter you from purchasing this sleek device a the low price fantastic screen and respectable battery life more than make up for it*" is shown in Fig. 2(*a*). As a side note, the errors in the sentence, and the following sentence examples, are caused by the lemmatizer (Sec. 3.1) that mistakenly recognizes "as" as a plural form and thus removes the "s" character from the word.

In Fig. 2(*a*), the direct relations are straightforward, i.e., for the adjectives "sleek," "fantastic," "respectable," and "low." For the adjective "cheap," since it has no nouns as either its direct parent or child, it moves toward the root. As it moves to the position of "feel" (word index 9), it obtains an indirect child "hand" (word index 15) and an indirect parent "plastic" (word index 6). Moving further to the position of "plastic" (word index 6), it obtains an indirect child "construction" (word index 4). Afterward, moving further until the root of the sentence does not generate any indirect child or parent. This tree becomes an example of various relations that are shown in Fig. 6. The search for indirect relations brings a trade-off, because it offers the possibility to obtain correct connections, e.g., "cheap" and "construction" in the example, but it is also likely to output false connections, e.g., "cheap" and "hand." Nevertheless, this paper keeps the indirect pairs.

For a sentence that contains a negation word, dependency tree helps to correctly relate the negation with the adjective which it negates. For example, there is a negation in the sentence "*the touch pad to move the cursor is not very good when i push it to click on a site*." The dependency tree for the sentence is shown in Fig. 2(*b*). Based on the tree, it can be determined that the word "not" negates the relation between "good" (word index 11) and "pad" (word index 3). The example in Fig. 2(*b*) also presents another advantage of using dependency tree compared to the distance-based approach in Refs. [8] and [11], i.e., the word "pad" is not adjacent with the adjective "good" in the sentence, yet the connection is correctly revealed by the tree.

4.2 Regression Results. Based on the connections obtained from the dependency tree, each pair of adjective and noun in a review sentence becomes a value that contributes to the corresponding variable in the regression model. The noun is interpreted based on its assignment to a product-feature word and the adjective is quantified based on its sentiment intensity. The existence of negation flips the sign of sentiment intensity. For example, as presented in Fig. 7, a review sentence for product i at time t that contains "sleek device" contributes as much as 0.853 to the variable "laptop+" for product i at time t. In order to reflect the effect of previous days' reviews toward the sales rank at time t, the contribution count is cumulated for the previous T time periods, i.e., $t, t-1, \ldots, t-T+1$.

There are 1990 data points for wearable technology products and 5587 data points for laptops. Data points included in the regression models must have the sales rank and price recorded for a particular date, as well as having reviews in the period within a week ($T=7$) from the date. In addition, the reviews must contain identified product-feature words along with the sentiment intensity. The regression analysis is done by applying stepwise regression to eliminate variables that are highly correlated with one another, using $\alpha = 0.05$. For the regression model of wearable technology products, the adjusted R-squared is 84.84% and the predicted R-squared is 84.23%. For laptops, the adjusted R-squared is 70.89% and the predicted R-squared is 70.33%, respectively. The significant independent variables ($\alpha = 0.05$) for both data sets are shown in Table 9.

5 Discussions

This section is divided into two parts. The first part analyzes the variables in the regression results and the second part assesses the sentence interpretation results as well as validating the proposed methodology.

5.1 Regression Result Analysis. First, it is worth noting from Table 9 that many textual-related variables are found to be significantly related to sales rank. It validates the inclusion of textual-related variables in the regression model. Interestingly, for both data sets, there are more significant variables from the review title than from the review content. For the data sets in the case study, this may suggest that a considerable number of customers pay most of their attention toward the review titles, and not reading the review content thoroughly.

Second, the coefficients confirm that the number of reviews and the percentage of reviews with good ratings (4 and 5 stars) are related to better sales rank. Accordingly, the percentage of reviews with bad ratings (1 and 2 stars) and higher price are related to worse sales rank. In the data sets, smaller number indicates better sales rank, i.e., rank 1 is better than rank 2.

Table 9 Regression results for wearable technology products and laptops

Wearable Technology Products			Laptops		
Variable	Coef	P-Value	Variable	Coef	P-Value
Constant	4.8076	0	Constant	4.9980	0
Price	0.0048	0	aveFractionVerified	0.1957	0.009
numReviews	−0.0092	0	numReviews	−0.0584	0
percent45stars	−0.2359	0.002	percent45stars	−0.3999	0
			percent12stars	0.2602	0.008
activity+	0.0393	0.007	apps-	0.0724	0.001
battery+	−0.0975	0.010	battery-	0.0756	0.033
charge-	0.1261	0	drive+	0.0702	0.021
company-support-service-	−0.0410	0	issue+	0.1368	0.001
data+	0.0421	0.031	laptop-	0.0430	0.008
data-	0.1603	0	life+	−0.1102	0.001
day-	−0.0661	0	office+	−0.1966	0
heartrate+	−0.0475	0.001	resolution-quality+	0.0440	0.009
problem+	−0.0656	0.011	screen-display+	−0.1030	0
			service-	−0.1978	0.001
			track-mouse+	−0.1744	0
			web-internet+	0.1748	0
			web-internet-	0.0968	0.002
title_activity-	−0.2420	0.009	title_apps-	−0.1534	0.028
title_alarm-	−0.5140	0	title_battery+	0.4199	0
title_battery-	−0.6650	0	title_battery-	0.2676	0.003
title_button+	1.7050	0.010	title_card+	−0.2916	0
title_button-	1.0600	0	title_card-	0.6340	0
title_charge-	−0.1917	0.002	title_drive+	−0.2790	0.012
title_company-support-service-	−0.0683	0.002	title_drive-	0.4530	0
title_data-	−0.2760	0.008	title_fan+	−1.2050	0
title_phone-laptop-app-	−0.1399	0.010	title_issue+	−0.2471	0.001
title_problem+	−0.1489	0.009	title_issue-	−0.1869	0.039
title_problem-	−0.1960	0	title_laptop-	−0.2006	0
title_screen+	−0.2300	0.034	title_office-	−1.6980	0
title_screen-	−0.7250	0	title_performance+	−0.0659	0.004
title_wristband-	0.1438	0	title_resolution-quality+	−0.1204	0.004
			title_service-	0.1781	0.033
			title_supervisor+	−0.0566	0.004
			title_supervisor-	0.1562	0

Third, among the significant textual-related variables, there are variables whose signs do not follow the common assumption. It is commonly assumed that a positive sentiment about a product feature (e.g., "battery+") is related to better sales rank, and vice versa. However, for example, the variable "activity+" in wearable technology products has a positive coefficient. Further observation reveals that the variable includes not only positive comments about the activity tracker but also positive comments about doing activity in general, e.g., "it make me more mindful of the exercise." The variable "(resolution-quality)+" in laptops also has a positive coefficient. The variable includes comments about sound quality, so terms such as "right speaker" and "left speaker" appear frequently. Due to the positive sentiment intensity for the adjectives, those neutral terms are interpreted as positive. As a result, it masks the actual complaint about the speaker in a sentence, e.g., "also my right speaker on the bad doesne very ." For the "(web-internet)+" variable, which also has a positive coefficient, further observation reveals that most of the sentences are interpreted correctly. However, many positive comments imply that the laptop only serves basic functions for internet, but it does not have capability to do more complicated tasks, e.g., "perfect for internet use not much else." Hence, the signs of the regression coefficients that do not follow the common assumption are explained.

An interesting finding is that the variables related to "problem" and "issues" have negative coefficients. It implies that, regardless of the sentiment intensity quantification (e.g., "major problem" is interpreted as "problem+"), the comments about problems are related to better sales rank. Further observation reveals that the word "deal" is assigned into the "problem" product-feature word and it contributes positive terms such as "great deal" and "real deal." Also, the statement of a problem may be followed by the positivity toward the product as a whole, e.g., "device has a couple issue but is okay especially since it is waterproofed and doesn't require frequent charging."

In the framework of the five-stage buying decision process, the significant variables in the regression models can suggest the pieces of information that are given significant weights by customers during the Evaluation of Alternatives stage. The information may be used by product designers as one of the inputs to improve product design. From the results shown in Table 9, the improvement efforts for wearable technology products may be considered for activity tracking functions, charging process, information presentation, quality and functions of the button, and the appearance of the product in general. The improvement efforts for laptops, as shown in Table 9, may be considered for nearly all aspects of a laptop, i.e., the applications, battery, storage space and memory, screen resolution, sound quality, and the quality of the laptop in general.

5.2 Assessment and Validation. To assess the interpretation of sentences from customer reviews, selected sentences from both data sets are presented in Table 10. The table provides an example of correctly interpreted sentence and three examples of falsely interpreted sentences for each data set. For the false interpretations, the source of the interpretation inaccuracy is indicated by the numbers inside the parentheses, i.e., (3.2) indicates the noncohesiveness of the group of words under a product-feature word, (3.3) indicates the inaccuracy of sentiment intensity score assigned to the adjective in the context of the given sentence, and (3.4) indicates the inability to capture the correct relation between

Table 10 Assesment of selected preprocessed review sentences

Assessment	Sentence (wearable technology products)	Adjective-Noun pair	Regression variable
True	work well short life span	"short span"	title_battery-
False (3.2)	it make me more mindful of the exercise i do during my day	"mindful exercise"	activity+
False (3.3)	dainty feminine long lasting battery	"lasting battery"	title_battery-
False (3.4)	work but need better quality control bought 2 only 1 is wearable battery lasted 2 week only	"wearable battery"	title_battery-
Assessment	Sentence (laptops)	Adjective-noun pair	regression variable
True	perfect for internet use not much else but based on price it 4 plus star	"perfect internet"	(web-internet)+
False (3.2)	like if yore skyping yo have purple dot all over	"purple dot"	(resolution-quality)+
False (3.3)	also my right speaker on the bad doesnt work	"right speaker"	(resolution-quality)+
False (3.4)	it is hard to get to where i want to go especially on the internet not that it is slow just hard to use	"slow internet"	(web-internet)-

an adjective and a noun. Those numbers correspond to the numbers of Methodology subsections in this paper.

The ideal validation would be comparing the results from the stages 3.2, 3.3, and 3.4 in Fig. 4 with a human-annotated corpus for both data sets. However, creating a reliable human-annotated corpus takes a considerable amount of time and effort. Moreover, it is hard to reach agreement between annotators for the tasks in this paper, e.g., an agreement on the set of relevant product features discussed in the reviews. Nevertheless, since all outputs from those stages build the regression models, the methodology is validated by the performance of the regression models, with predicted R-squared as the performance measure.

The predicted R-squared values are obtained high for both models and they do not drop drastically from the adjusted R-squared values. Thus, it can be concluded that, despite the inaccuracies in the natural language processing, the regression models provide a good description of the relation between reviews and sales rank and they would generalize well to a new data set. It is worth noting that predicting sales rank accurately is not a main purpose of this paper. Therefore, prediction accuracy is not used as a performance measure.

6 Conclusions and Future Works

The paper proposes a methodology to identify the relation between online customer reviews and sales rank. The methodology consists of five main stages, i.e., data preprocessing, product-feature words identification, sentiment intensity quantification, dependency tree interpretation, and regression model generation. The methodology involves minimal subjective inputs, rules, and decisions such that the model is objective and generalizable into a new data set. The methodology reveals the product features that are significantly related to sales rank.

The methodology is applied to two data sets, i.e., wearable technology and laptop products. For both data sets, the performance of the regression models is good, i.e., the predicted R-squared is 84.23% for wearable technology products and 70.33% for laptops. The high predicted R-squared values support the claim that the model is generalizable.

For future works, to improve the accuracy of interpreting customer reviews, a better word embedding can be achieved by applying word sense disambiguation [26] to a word that has different meanings. In the case of wearable technology products, for example, the word "charge" means either refilling a battery by passing a current through it or the name of a product variant from Fitbit. Also, an improved method is required to determine the correct connections between a pair of product-feature and sentiment words from the dependency tree.

Nomenclature

$aLen_{i,T}$ = average length of reviews for product i during period T
$aRat_{i,T}$ = average rating of reviews for product i during period T
$aVer_{i,T}$ = average number of verified purchase of product i during period T
$BIC(M_j)$ = the BIC value of the jth model in X-means clustering
D = set of review documents
$fN_{f,i,T}$ = count of negative comments of feature f for product i during period T
$fP_{f,i,T}$ = count of positive comments of feature f for product i during period T
idf_i = inverse document frequency of word i
$int(w)$ = sentiment intensity of word w
$\hat{l}_j(D_{\mathrm{BIC}})$ = log likelihood of data D in BIC computation
$nRev_{i,T}$ = number of reviews posted for product i during period T
p_j = number of parameters in the j-th model in X-means clustering
$pF_{i,T}$ = fraction of reviews for product i during period T that are rated 4 and 5 stars
$pO_{i,T}$ = fraction of reviews for product i during period T that are rated 1 and 2 stars
$Price_{i,t}$ = price of item i at time t
R = number of data points in data D in BIC computation
$Rank_{i,t}$ = rank of product i at time t
S_w = set of words that have the highest cosine similarity with word w
$sim(w,s)$ = cosine similarity between word w and word s
T = number of time periods within the bracket of time $(t - T)$ and t
tf_i = term frequency of word i, i.e. count of word i
$tFN_{f,i,T}$ = count of negative comments of feature f in the title review for product i during period T
$tFP_{f,i,T}$ = count of positive comments of feature f in the title review for product i during period T
ti_i = tf.idf value of word i
$u_{i,j}$ = the value of element j at the i-th output vector
ν_i = fixed-effect variable for product i
w_i = word i in the vocabulary
y_i = data point i
$\hat{y}_{i,-i}$ = fitted value for data point i in the model that omitting i
$y_{i,j}$ = probability of word j being a context word at the i-th output vector

References

[1] Kotler, P., and Keller, K. L., 2006, *Marketing Management*, 12th ed., Pearson Education, Upper Saddle River, NJ, pp. 184–199.
[2] Decker, R., and Trusov, M., 2010, "Estimating Aggregate Consumer Preferences From Online Product Reviews," Int. J. Res. Mark., **27**(4), pp. 293–307.
[3] Chevalier, J. A., and Mayzlin, D., 2006, "The Effect of Word of Mouth on Sales: Online Book Reviews," J. Mark. Res., **43**(3), pp. 345–354.
[4] Sun, M., 2012, "How Does the Variance of Product Ratings Matter?," Manage. Sci., **58**(4), pp. 696–707.

[5] Jindal, N., and Liu, B., 2007, "Review Spam Detection," 16th International Conference on World Wide Web (WWW), Banff, AB, Canada, May 8–12, pp. 1189–1190.
[6] Guo, H., Zhu, H., Guo, Z., Zhang, X., and Su, Z., 2009, "Product Feature Categorization With Multilevel Latent Semantic Association," 18th ACM Conference on Information and Knowledge Management (CIKM '09), Hong Kong, China, Nov. 2–6, pp. 1087–1096.
[7] Suryadi, D., and Kim, H. M., 2017, "Identifying Sentiment-Dependent Product Features From Online Reviews," *Design Computing and Cognition '16*, J. S. Gero, ed., Springer International Publishing, Basel, Switzerland, pp. 685–701.
[8] Suryadi, D., and Kim, H., 2016, "Identifying the Relations Between Product Features and Sales Rank From Online Reviews," ASME Paper No. DETC2016-60481.
[9] Chong, A. Y. L., Ch'ng, E., Liu, M. J., and Li, B., 2017, "Predicting Consumer Product Demands Via Big Data: The Roles of Online Promotional Marketing and Online Reviews," Int. J. Prod. Res., **55**(17), pp. 5142–5156.
[10] Quan, X., Wenyin, L., and Dou, W., 2011, "Longitudinal Sales Responses With Online Reviews," 11th IEEE International Conference on Data Mining Workshops (ICDMW), Vancouver, BC, Dec. 11, pp. 103–108.
[11] Hu, M., and Liu, B., 2004, "Mining Opinion Features in Customer Reviews," 19th National Conference on Artifical Intelligence, San Jose, CA, July 25–29, pp. 755–760.
[12] Hu, M., and Liu, B., 2004, "Mining and Summarizing Customer Reviews," Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (KDD '04), Seattle, WA, Aug. 22–25, pp. 168–177.
[13] Wei, C.-P., Chen, Y.-M., Yang, C.-S., and Yang, C. C., 2010, "Understanding What Concerns Consumers: A Semantic Approach to Product Feature Extraction From Consumer Reviews," Inf. Syst. e-Bus. Manage., **8**(2), pp. 149–167.
[14] Abulaish, M., Jahiruddin, Doja, M. N., and Ahmad, T., 2009, *Feature and Opinion Mining for Customer Review Summarization*, Springer, Berlin, Heidelberg, pp. 219–224.
[15] Ma, B., Zhang, D., Yan, Z., and Kim, T., 2013, "An LDA and Synonym Lexicon Based Approach to Product Feature Extraction From Online Consumer Product Reviews," J. Electron. Commerce, **14**(4), pp. 304–314.
[16] Mei, Q., Ling, X., Wondra, M., Su, H., and Zhai, C., 2007, "Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs," 16th International Conference on World Wide Web (WWW '07), Banff, AB, Canada, May 8–12, pp. 171–180.
[17] Archak, N., Ghose, A., and Ipeirotis, P. G., 2007, "Show Me the Money!: Deriving the Pricing Power of Product Features by Mining Consumer Reviews," 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (KDD '07), San Jose, CA, Aug. 12–15, pp. 56–65.
[18] Kobayashi, N., Inui, K., and Matsumoto, Y., 2007, "Extracting Aspect-Evaluation and Aspect-of Relations in Opinion Mining," Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague, pp. 1065–1074.
[19] Netzer, O., Feldman, R., Goldenberg, J., and Fresko, M., 2012, "Mine Your Own Business: Market-Structure Surveillance Through Text Mining," Mark. Sci., **31**(3), pp. 521–543.
[20] Archak, N., Ghose, A., and Ipeirotis, P. G., 2011, "Deriving the Pricing Power of Product Features by Mining Consumer Reviews," Manage. Sci., **57**(8), pp. 1485–1509.
[21] Zhou, F., Jiao, R. J., and Linsey, J. S., 2015, "Latent Customer Needs Elicitation by Use Case Analogical Reasoning From Sentiment Analysis of Online Product Reviews," ASME J. Mech. Des., **137**(7), p. 0714011.
[22] Somprasertsri, G., and Lalitrojwong, P., 2008, "Automatic Product Feature Extraction From Online Product Reviews Using Maximum Entropy With Lexical and Syntactic Features," IEEE International Conference on Information Reuse and Integration, Las Vegas, NV, July 13–15, pp. 250–255.
[23] Somprasertsri, G., 2010, "Mining Feature-Opinion in Online Customer Reviews for Opinion Summarization," J. Universal Comput. Sci., **16**(6), pp. 938–955.
[24] Jin, W., Ho, H. H., and Srihari, R. K., 2009, "Opinionminer: A Novel Machine Learning System for Web Opinion Mining and Extraction," 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Paris, France, June 28–July 1, pp. 1195–1204.
[25] Zhai, Z., Liu, B., Xu, H., and Jia, P., 2011, "Clustering Product Features for Opinion Mining," Fourth ACM International Conference on Web Search and Data Mining (WSDM '11), Hong Kong, China, Feb. 9–12, pp. 347–354.
[26] Jurafsky, D., and Martin, J. H., 2009, *Speech and Language Processing*, 2nd ed., Pearson Education, Upper Saddle River, NJ.
[27] Cambria, E., Poria, S., Bajpai, R., and Schuller, B. W., 2016, "Senticnet 4: A Semantic Resource for Sentiment Analysis Based on Conceptual Primitives," 26th International Conference on Computational Linguistics (COLING), Osaka, Japan, Dec. 11–16, pp. 2666–2677.
[28] Kim, S.-M., and Hovy, E., 2006, "Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text," Workshop on Sentiment and Subjectivity in Text (SST '06), Sydney, Australia, July 22,, Association for Computational Linguistics, pp. 1–8.
[29] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J., 2013, "Distributed Representations of Words and Phrases and Their Compositionality," e-print arXiv:1310.4546.
[30] Zhang, D., Xu, H., Su, Z., and Xu, Y., 2015, "Chinese Comments Sentiment Classification Based on word2vec and SVMperf," Expert Syst. Appl., **42**(4), pp. 1857–1863.
[31] Levy, O., and Goldberg, Y., 2014, "Dependency-Based Word Embeddings," 52nd Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Baltimore, MD, June 23–25, pp. 302–308.
[32] Rong, X., 2014, "word2vec Parameter Learning Explained," e-print arXiv:1411.2738
[33] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D., 2014, "The Stanford CoreNLP natural language processing toolkit," Association for Computational Linguistics (ACL) System Demonstrations, Baltimore, MD, pp. 55–60.
[34] Řehůřek, R., and Sojka, P., 2010, "Software Framework for Topic Modelling With Large Corpora," LREC 2010 Workshop on New Challenges for NLP Frameworks, La Valleta, Malta, May 22, pp. 45–50.
[35] Pelleg, D., and Moore, A. W., 2000, "X-Means: Extending k-Means With Efficient Estimation of the Number of Clusters," 17th International Conference on Machine Learning, ICML '00, San Francisco, CA, June 29–July 2, pp. 727–734.
[36] Quan, C., and Ren, F., 2014, "Unsupervised Product Feature Extraction for Feature-Oriented Opinion Determination," Inf. Sci., **272**, pp. 16–28.
[37] Cambria, E., Hussain, A., Havasi, C., and Eckl, C., 2009, "Affectivespace: Blending Common Sense and Affective Knowledge to Perform Emotive Reasoning," WOMSA09, Seville, Spain, pp. 32–41.
[38] Forman, C., Ghose, A., and Wiesenfeld, B., 2008, "Examining the Relationship Between Reviews and Sales: The Role of Reviewer Identity Disclosure in Electronic Markets," Inf. Syst. Res., **19**(3), pp. 291–313.
[39] Miller, A., 2002, *Subset Selection in Regression*, 2nd ed., Chapman & Hall/CRC, Boca Raton, FL.