Journal of Mechanical Design. Received April 26, 2021;
Accepted manuscript posted October 15, 2021. doi:10.1115/1.4052904

Phrase Embedding and Clustering for Sub-Feature Extraction from Online Data

Seyoung Park, Harrison M. Kim*

Enterprise Systems Optimization Laboratory
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801
Email: seyoung7@illinois.edu, hmkim@illinois.edu

Recently, online user-generated data has been used as an efficient resource for customer analysis. In the product design area, various methods for analyzing customer preference for product features have been suggested. However, most of them focused on feature categories rather than product components which are crucial in practical applications. To address that limitation, this paper proposes a new methodology for extracting sub-features from online data. First, the method detects phrases in the data and filtered them using product manual documents. The filtered phrases are embedded into vectors, and then they are divided into several groups by two clustering methods. The resulting clusters are labeled by analyzing items in each cluster. Finally, cue phrases for sub-features are obtained by selecting clusters with labels representing product features. The proposed methodology was tested on smartphone review data. The result provides feature clusters containing sub-feature phrases with high accuracy. The obtained cue phrases will be used in analyzing customer preferences for sub-features and this can help product designers determine the optimal component configuration in embodiment design.
Keywords: data mining, feature extraction, online data

1 Introduction

In the manufacturing industry, product launch consists of several tasks, including product design, marketing planning, product development, quality examination, assembly, and production [1]. It is a cyclic process due to engineering changes such as physical forms, materials, and functions [2]. These changes result in additional costs in product launch, so companies make great efforts to establish the initial product design as accurately as possible. They collect various types of data such as market trends, technology trends, and customer opinions and draw implications for product design. Regarding customer opinions, the conventional data sources are customer surveys, group interviews, and expert interviews. However, these methods have a limitation in that they require much time and cost. Also, the answers may be biased by the incorrectly designed questionnaires. As an alternative, online user-generated data has been drawing attention. With the development of various online channels and communication devices, online customer opinions on products have increased exponentially. This data is mainly generated in open media such as social networking services and online shopping websites, so the data is always accessible. This makes collecting customer opinions faster and easier than traditional methods.

Various studies have been conducted on online customer data analysis. Some of them focused on numeric parts of the data including the number of reviews and ratings [3, 4, 5]. Others focused on textual data analysis using Natural Language Processing (NLP) techniques such as association mining [6], Term Frequency and Inverse Document Frequency (TF-IDF) [7,8], Latent Dirichlet allocation (LDA) [9,10,11], and Word2vec [12]. This paper focuses on textual data analysis for extracting customer opinions on product features. Most research in this area started with extracting product features from online data. Various methods for feature extraction were suggested [13, 14, 15], using different NLP techniques. However, they have limitations in terms of practical application because they extract feature categories rather than specific sub-features.

In the industrial field, product development requires embodiment design, which can be defined as the physical form such as product architecture, modeling of parts, and final product dimensions [16]. Since a product is manufactured by configuring multiple components, a general feature consists of several sub-features. For example, in smartphones, the camera feature includes two parts - rear and front camera modules. Also, a part is described by its features. Specifically, screen components have different sizes, resolutions, and types. In this paper, the term 'sub-feature' means both the part and part features. These sub-features are crucial factors in the embodiment design. But most studies extract main

*Corresponding Author

features and analyze them instead of sub-features.

This research aims to address the above limitation and provide a solution for embodiment design. The proposed methodology is an extension of the previous work by Suryadi & Kim [15]. They presented a method for extracting feature terms from online data sing a word2vec model [12]. In their method, words in the review data are embedded into a vector space and then these word vectors are grouped into different clusters. By filtering cluster center words with product manuals, it identifies a set of product feature clusters. Although this method provides an automated way of identifying and grouping feature words, it has some limitations. First, the method considers nouns only, making it hard for sub-features to be detected. For example, the method can detect 'camera' but it can hardly detect the subdivided features of 'camera' such as 'rear camera' and 'front camera'. Second, the review data itself contains many noise words. As a result, a large portion of data points in the vector space is noise. The original paper also pointed out that not all nouns are equally significant. To reflect the importance of each noun, the methodology put a weight on each data point according to its TF-IDF. However, it did not solve the inefficient clustering caused by noise. Moreover, the methodology reassigns nouns in non-feature clusters into feature clusters. In some ways, this process is the reverse of clustering and it lowers the accuracy of the items in feature clusters. This paper addresses the above-mentioned limitations with a new methodology based on phrase embedding and clustering. It expands the range of extracted features by considering phrase vectors instead of noun vectors. Also, the proposed methodology applies new clustering methods which are more effective in synonym extraction than K-means clustering.

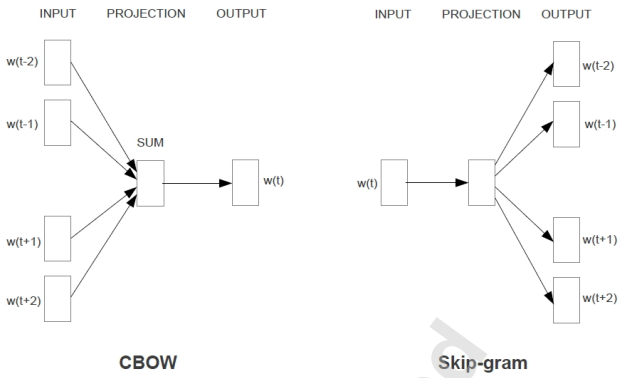
The rest of the paper is organized as follows. In section 2, relevant literature will be introduced. In section 3, the detailed process of the proposed methodology will be explained. Section 4 will show the results of the new method conducted on online review data. In Section 5, the simulation result will be evaluated. The improvements will be demonstrated by comparing the result of the new and the previous methodology for the same data. Finally, in section 6, findings will be summarized, and future works will be discussed.

2 Literature Review

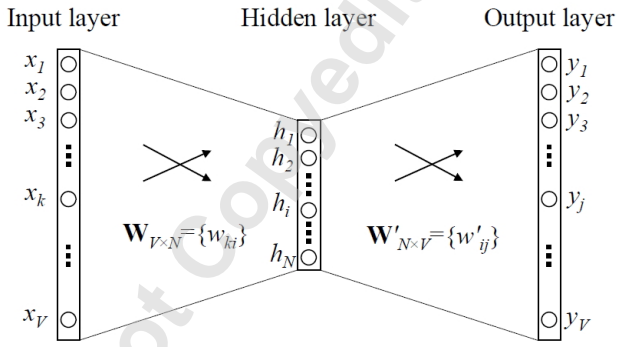
2.1 Word2vec

The word2vec model introduced by Mikolov et al. [12] [17] is one of the well-known Natural Language Processing (NLP) models. It provides distributed representations of words learned by neural networks. As shown in Fig. 1a, the model has two kinds of architectures, a Continuous Bag-of-Word (CBOW) and a skip-gram. In a CBOW model, the target output is a certain word in a sentence and the input data is the words surrounding that target word. The model is trained to learn word vector representations that are good at predicting this target word. A skip-gram model uses the opposite architecture.

The detailed training process is explained by Rong [18]. Fig. 1b shows a simple CBOW model with one input word



(a) Architectures [17]



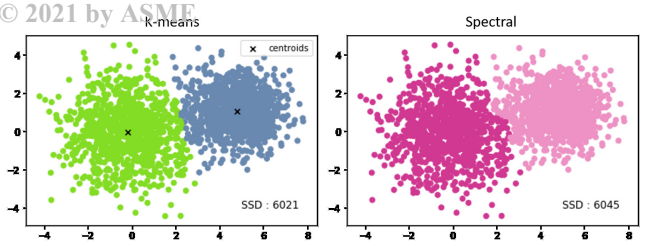
(b) Training [18]

Fig. 1: Word2Vec Model

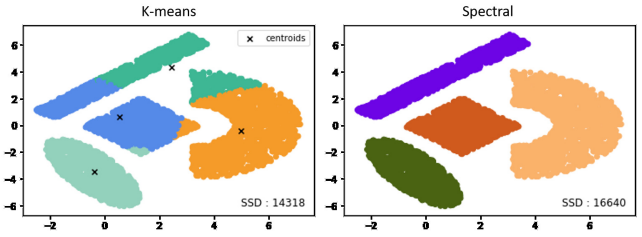
and one output word. The input layer is a V-dimensional one-hot encoded vector where only one element has a value 1 and all others are 0. The hidden layer is an N-dimensional vector. The weights between these two layers can be represented by a matrix $W_{V\times N}$. Since the input is a one-hot encoded vector, the hidden layer works as a projection layer. To be specific, the input for k^{th}word is the vector where x_k is 1, and the rests are 0. Then, $W^T X$ results in k^{th} row of W, which is essentially copying the k^{th} row of W to the hidden layer h. From the hidden layer to the output layer, there is another weight matrix $W'_{N\times V}$. The vector $U = W'^T h$ is the input vector for the output layer. Let $u_j = v'^T_{w_j} h$ where v'_{w_j} is the j^{th} column of W'. Then u_j is the input value for the j^{th} unit in the output layer. Unlike the hidden layer, the output layer units have an activation function. Here, softmax is used.

$$\begin{aligned} \frac{\partial E}{\partial w'_{ij}} &= \frac{\partial E}{\partial u_j} \cdot \frac{\partial u_j}{\partial w'_{ij}} = e_j \cdot h_i \\ \frac{\partial E}{\partial w_{ki}} &= \frac{\partial E}{\partial h_i} \cdot \frac{\partial h_i}{\partial w_{ki}} = (\sum_{j=1}^{V} \frac{\partial E}{\partial u_j} \cdot \frac{\partial u_j}{\partial h_i}) \cdot x_k \end{aligned} \tag{1}$$

The next step is the back-propagation process. In this step, the weight matrices W and W' are updated to minimize the loss function E of the network. The update equations in Eq. 1 are derived using the chain rule of derivatives and the



(a) Gaussian Data



(b) Shaped Data

Fig. 2: K-means vs. Spectral Clustering

stochastic gradient descent. For the weights between hidden and output layers, we take the derivative of E with respect to w'_{ij}. For the weights between input and hidden layers, we first take the derivative of E with respect to h_i, and then compute the update equation for w_{ki}. The training iterates until E meets a certain criterion.

2.2 Clustering

Clustering is a method for grouping objects according to measured or perceived intrinsic characteristics of similarity [19]. Traditional clustering can be divided into many categories [20]. In this paper, clustering algorithms based on a partition, graph theory, and density will be discussed. Partition-based clustering regards the center of data points as the center of a corresponding cluster. A representative method is K-means clustering [21] based on Lloyd's algorithm [22]. The objective of the algorithm is to minimize the cost which is defined as the sum of squared distances between data points and their cluster centers. K-means clustering works well on the dataset with mixed Gaussian distribution. Clustering methods based on graph theory regard the total dataset as a graph. Each data point is a node and the relationship among data points defines edges. Spectral clustering [23] [24] is one of the typical algorithms in graph theory-based clustering. In spectral clustering, the pairwise similarity of data points is measured and a proper matrix is derived from this measured similarity. Then, eigenvalues and eigenvectors of this matrix are used to divide data points. Spectral clustering performs well in shaped data. Fig. 2 shows the results of K-means and spectral clustering for the different types of data. For a simple mixture of Gaussian shown in Fig. 2a, both techniques work well. However, for the shaped data in Fig. 2b, spectral clustering distinguishes four different shapes in the data while K-means clustering does not. Density-based clustering is a relatively recently proposed clustering method. One of the state-of-the-art techniques is HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) [25]. It builds a minimum spanning tree (MST) based on the mutual reachability distance among data points. Then, a cluster hierarchy is created by merging edges in the MST. This cluster tree is condensed so that each cluster is larger than the minimum cluster size. The algorithm performs well for the data with different densities.

2.3 Feature Extraction from Online Data

In data-driven design, various approaches for feature extraction have been suggested. Joung & Kim [13] adopt LDA to extract feature words from online data. They analyze topics in customer reviews using LDA, and it returns sets of a topic and relevant words. The authors select feature-related topics and extract words belonging these topics. Tuarob & Tucker [14] utilizes a bootstrapping algorithm to detect feature-related terms in Twitter mentions. They initially present a set of ground-truth features. Then, the algorithm repeatedly learns phrase templates surrounding the seed features. It detects similar sentence patterns and identifies feature words with noun POS tagging. Zhang et al. [26] extract feature synonyms from Amazon reviews using Word2vec. They prepare seed words for features and calculate the semantical distances between seed words and other words based on cosine similarity. The higher cosine value means a closer semantic relationship. The authors sort the similarity scores and select the list of words closest to the seed word.

Suryadi & Kim [15] propose a methodology using Word2vec and clustering, on which this study is based. First, laptop reviews are collected from Amazon.com, and then review sentences are cleaned and analyzed by an NLP toolkit in PYTHON. The output contains several pieces of information for words in a review sentence. They are lemmatized word formats, Part of Speech(POS) tags, and a dependency tree that represents the relationship among words in the sentence. Next, lemmatized words are trained by the word2vec model and embedded into the vector distribution. Suryadi & Kim assume that feature words would be generally nouns, so they filter word vectors with noun POS tags only. These filtered noun vectors are grouped by X-means clustering [27], an extended K-means that automatically determines the number of clusters based on BIC scores. It is assumed that relevant words are located closer than irrelevant words. Therefore, clustering would bring similar words into the same group. After clustering, the word closest to the cluster center becomes the center word. Among these center words, feature words are chosen by using product manual documents. The center words with the frequency above a certain threshold are selected as feature words, and clusters to which feature words belong become feature clusters. Finally, clustering is finalized by assigning words in non-feature clusters to the closest feature cluster.

These studies identify feature categories mentioned in online data. However, they cannot extract sub-features which are crucial in embodiment design. The new methodology presented in the following section addresses this limitation.

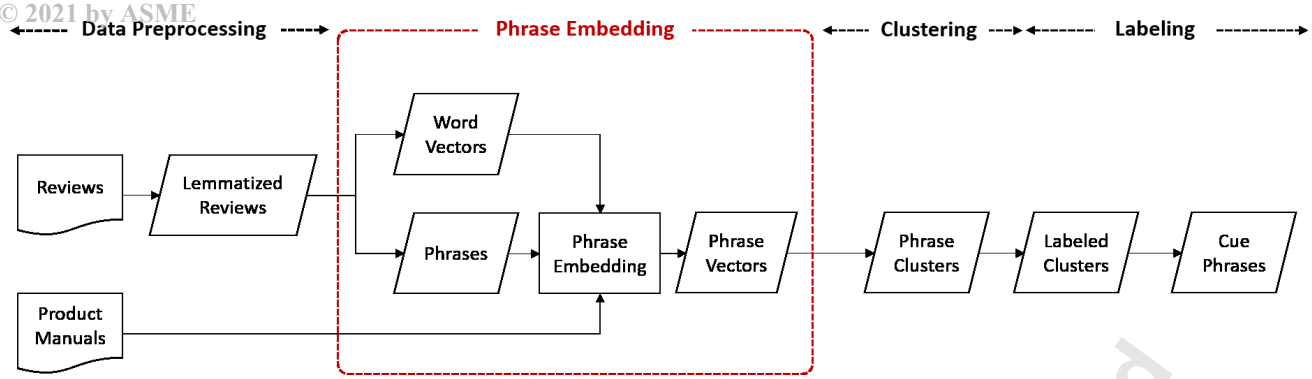


Fig. 3: Flowchart of The Proposed Methodology

3 Methodology

Fig. 3 shows the overall process of the proposed methodology which consists of four stages. In the first stage, online data is collected, cleaned, and lemmatized. In the phrase embedding stage, lemmatized words from the previous stage are vectorized by Word2Vec. Then, phrases are extracted from the data and embedded into a vector space using word vectors and product manuals. In the next stage, the phrase vectors are clustered into several groups. In the last stage, each cluster is labeled by the Term-Frequency (TF) analysis, and cue phrases for product features are extracted.

3.1 Data Preprocessing

In this stage, two types of data are collected: (i) online user-generated data (e.g., online reviews); (ii) product manual documents distributed by manufacturers. The former is for feature extraction and the latter is used in phrase embedding. The collected datasets are free-format text contents, so they are cleaned to improve the performance of word embedding. In this research, non-letters including special characters and punctuation marks are removed from the data. Also, all uppercase letters are converted to lowercase. Stopwords and non-English words are not removed because it degrades the result of phrase extraction to be performed in the next stage. Moreover, removing non-English words will exclude feature-related words such as 'GB' (unit for memory size) and 'mAh' (unit for battery capacity). After the cleaning process, the review sentences are analyzed by an NLP toolkit that provides linguistic characteristics of each word. Among various characteristics, the lemmatized form and POS tagging are used in this study.

3.2 Phrase Embedding

This stage consists of three steps: (i) word embedding; (ii) phrase extraction; (iii) phrase embedding.

First, in the word embedding step, all words from the online data are embedded into vectors using Word2Vec [12]. In the previous study [15], noun-vectors are filtered by checking the POS tag of each word. In this study, the filtering process is removed because sub-features can be represented by non-noun phrases. For example, a customer may express screen size by 'large screen' or 'small screen'.

Next, phrases are extracted from the online data by the NLP toolkit. Since this study aims to extract cue phrases for product features, feature-irrelevant phrases such as 'birthday present' and 'school work' are considered as noise. These phrases can be removed by using product manual documents. The previous methodology [15] used a one-sample T-test to filter out cluster center words in the final stage. This study removes noise words in advance so that the performance of clustering can be improved. To increase the diversity of extracted features, word frequency is used as the criterion for noise filtering instead of T-test. If any word in a phrase does not appear in the product documents, then the phrase is assumed to be irrelevant to the target product and removed. There are other types of noise phrases. Brand names and product categories are included in the product manuals but they are not related to product features. Therefore, by designating certain words as noise, phrases such as 'Samsung phone' and 'Apple smartphone' can be filtered out.

Finally, the selected phrases are embedded into a vector space based on Eq. 2 where $\vec{W}_i$ is the vector representation of word i.

$$Phrase = a_1\vec{W}_1 + a_2\vec{W}_2$$
$$a_1 = \frac{F(W_1)}{F(W_1)+F(W_2)} \quad a_2 = \frac{F(W_2)}{F(W_1)+F(W_2)} \tag{2}$$

The weight a_i represents the importance of each word in terms of product features. It is calculated by the ratio of word frequencies in the product manual documents. The assumption here is that when a phrase is composed of multiple words, the word with the higher frequency in product manuals is more likely to be a product feature than those with lower frequencies. This assumption is tested and verified by analyzing product manual documents. The analysis result is shown in Fig. 4 and the details will be explained in Section 4. This approach can be applied to phrases with more than two words. Since the phrase has a hierarchy with the main feature, sub-feature, and sub-sub-features, the weighted sum of word vectors will be closer to the vector of the main feature. In this study, for the simplicity of phrase embedding, only phrases consisting of two words are selected and used.

3.3 Phrase Clustering

In this stage, the embedded phrases are clustered. The previous study [15] used X-means clustering which is based on the distance between each centroid and each data point. However, Zhang et al. [28] showed that spectral clustering based on the pairwise similarity of data points provides much more accurate synonym groups than X-means clustering. One limitation of spectral clustering is that the number of clusters should be set manually. To solve this problem, this study uses two types of clustering methods - HDBSCAN and spectral clustering. The phrase vectors are used without normalization. First, HDBSCAN, a density-based clustering method, is applied to the phrase vectors, and it automatically determines the optimal number of clusters (K). Next, spectral clustering is conducted for the same data with this optimal K. In the result, two sets of clustering results are obtained, and both are analyzed by cluster labeling and then merged. In this way, the downside of HDBSCAN, exclusion of outlier phrases, can be compensated by spectral clustering that includes all items in the result. This study groups synonym phrases in an automated way with high accuracy by combining two clustering methods.

3.4 Cluster Labeling

The resulting clusters from the previous stage can be labeled by analyzing items in each cluster. The previous study [15] labeled a cluster by extracting its center word. In this study, the cluster label was determined by counting term frequencies in each cluster. Since all phrases are embedded by Eq. 2, the phrase vectors close to each other share the same word. As a result, most phrases in the same cluster have a common word. Therefore, a word with the highest frequency in a cluster would be a subject of that cluster. For illustration, let us assume one of the clusters contains 'screen size', 'screen resolution', 'screen brightness', and 'screen ratio'. The manual analysis will label this cluster as 'screen'. The frequencies of words are {screen: 4, size: 1, resolution: 1, brightness: 1, ratio: 1} and the word with the highest frequency is 'screen' which is the cluster's subject. This shows that TF analysis can be an efficient method for identifying each cluster's topic. For TF analysis, phrases in the cluster are broken down into words and the frequency of each word is counted. Then, every cluster is labeled with the most frequent word within it. At the end of this stage, pairs of a subject label and related cue phrases are obtained

Since these cue phrases represent sub-features, product designers can draw practical design implications by utilizing them. For example, the designers can detect sentences mentioning specific sub-features. Then, they can analyze sentiments for those sub-features and determine which ones to improve within a limited budget.

4 Case Study

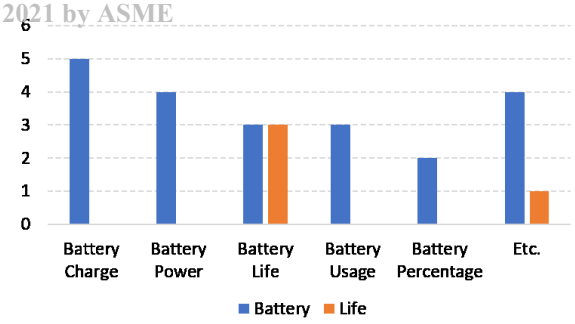
4.1 Data Preprocessing

The proposed methodology aims to extract product sub-features from online data. This paper selected smartphones for the case study for two reasons. First, a smartphone is a highly integrated electronic device, so most smartphone features consist of multiple sub-features. Second, most people in the US are familiar with product features with an 85% penetration rate [29]. The methodology requires two types of data: (i) online user-generated data; (ii) product manual documents. For (i), the smartphone review data was collected from Amazon.com but the data source is not limited to the online shopping websites. Other online platforms providing free-format text data can be used as the source of user-generated data. For the collected data, the total number of reviews is 25,340 for 58 products and the reviews are written from May 2017 to July 2020. For the authenticity of the data, only the reviews marked as 'verified purchase' by Amazon were used. The review data contains 109,688 sentences and 18,419 unique words. Each review has 4 sentences with 43 words in average. The product manuals are documents distributed online by manufacturers. Specifically, manual documents for six different smartphones were used: Samsung Galaxy fold, Galaxy S10, Apple iPhone, OnePlus 7T, Xiaomi Mi, ZTE Blade Z Max. The collected datasets were cleaned and lemmatized using Spacy library in PYTHON. In specific, special characters are removed, and all punctuations are replaced with a period. Upper case letters are transformed into lower cases, and all words are lemmatized. Stopwords are not removed as it affects the phrase extraction to be performed in the next stage.

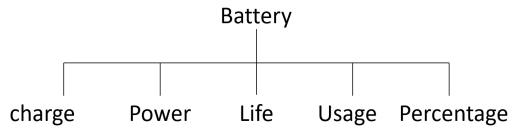
4.2 Phrase Embedding

As mentioned in Section 3, phrase clustering consists of word embedding, phrase extraction, and phrase clustering. In the word embedding step, Gensim library in PYTHON was used to conduct Word2Vec modeling. Regarding parameters, the same values as in the previous study [15] were used. The dimension of the vector is 100, the number of windows is 2, and the minimum word count is 8. After training, Gensim returned a set of word vectors. For phrase extraction, Spacy library in PYTHON was used. Spacy provides two different methods for phrase extraction - Noun_chunk and Textrank. Some of the phrases extracted from the two methods were common, but there were also non-common phrases. This study used both methods and combined the results to increase the amount of extracted phrases. These phrases were filtered by the following conditions: (i) The frequency of a phrase in review data is greater than 4; (ii) At least one word in a phrase should appear in product manuals; (iii) Both words are not noise terms. In this case study, the predefined noise terms were [product, phone, samsung, apple, google, iphone, galaxy, day, week, month, year, time, -PRON-, -pron-]. As a result, 1,302 phrases were obtained.

The filtered phrases were embedded into a vector space by Eq. 2. The equation is based on the assumption that words that appear more frequently in product manuals are more likely to be main features than words with lower frequencies. To validate this assumption, the product manual document was analyzed. Fig. 4a shows the analysis result for 'battery life'. In a smartphone manual, 'battery' appears 33 times



(a) Context



(b) Word Hierarchy

Fig. 4: Product Manual Analysis

and 'life' appears 4 times. While 'battery' is used in several feature contexts (battery charge, battery power, battery life, battery usage, and battery percentage), 'life' is used in only one feature context (battery life). These contexts imply a hierarchy shown in Fig. 4b where 'battery life' is a sub-feature of 'battery'. Considering both frequency and hierarchy, we can conclude that the phrase 'battery life' belongs to a main feature 'battery' which has a higher frequency than 'life'. Therefore, Eq. 2 will assign a proper vector point to each phrase. Fig. 5 shows the distribution of noun vectors and phrase vectors using Principal Component Analysis (PCA) [30]. 1,217 noun vectors from 25,340 reviews are shown in Fig. 5a and 1,302 phrase vectors from the same data are shown in Fig. 5b and 5c. The phrase vectors are scattered while the noun vectors are concentrated in the center.

4.3 Phrase Clustering

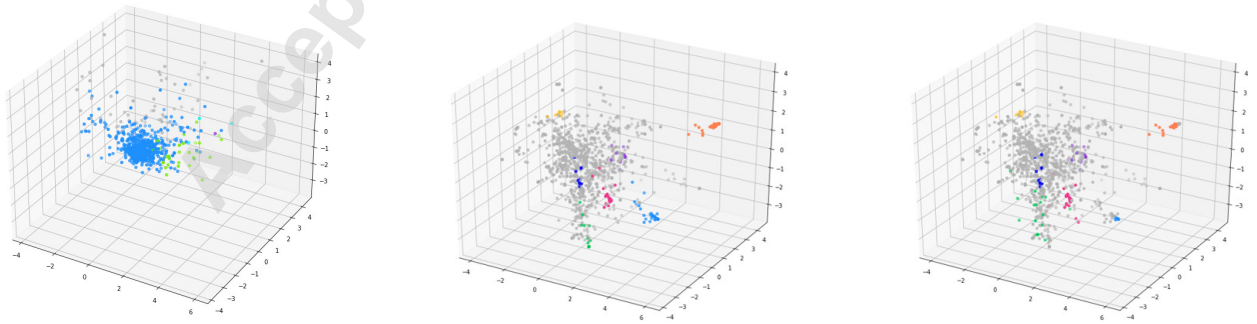
The previous study [15] applied X-means clustering for noun vectors. It was tested on the smartphone review data using pyclustering library in PYTHON. In Fig. 5a, feature-related clusters are marked in colors. As shown in the figure, X-means clustering produced one big feature cluster on the center and several small feature clusters around the edges. The big cluster contains many noise words and the details will be discussed in Section 5. The new methodology presented in this paper applied HDBSCAN and spectral clustering for phrase vectors. In the case study, hdbscan library in PYTHON was used for HDBSCAN and scikit-learn library in PYTHON was used for spectral clustering. First, HDBSCAN was applied to the phrase vectors and automatically produced 72 clusters. Then, with $K = 72$, spectral clustering was conducted for the same phrase vectors. Fig. 5b and Fig. 5c visualize the distribution of feature-related clusters from HDBSCAN and spectral clustering respectively.

The clustering results can be evaluated by a numerical index. Considering that the purpose of this study is to group synonyms, the Davies-Bouldin (DB) index [31] would be an appropriate criterion. The DB index measures the average similarity between each cluster and its most similar one. Since it is desirable for the clusters to have the minimum similarity to each other, the lower score represents the better clustering. In the previous method with nouns, the DB score of X-means clustering is 1.453. In the new method with phrases, spectral clustering provides the DB score of 1.239, and HDBSCAN results in 0.516. The proposed method provides better results in terms of feature clustering.

The results from HDBSCAN and spectral clustering were not the same due to the difference in algorithms. Table 1 shows the items in a certain cluster from two results. The items are related to the security feature of the smartphone. Some items appear in both results but there are some uncommon items that are marked in boldface. In specific, the cluster from HDBSCAN contains phrases relevant to fingerprint and iris. The result of spectral clustering contains not only those phrases but also phrases about face recognition. These two clustering results were merged into one cluster to enhance the diversity of cue phrases for each topic.

4.4 Cluster Labeling

The resulting clusters were analyzed by term frequency (TF) [7] so that each cluster can be labeled by its topic. For



(a) Noun Only Clustering (b) Phrase Embedded Clustering (H) (c) Phrase Embedded Clustering (S)

Fig. 5: Vector Distributions

Table 1: Clusters Related to The Security Feature

HDBSCAN	finger scanner, finger reader, finger sensor, fingerprint reader, iris scanner, fingerprint sensor, finger print, fingerprint scanner, fingerprint reading, fingerprint recognition, same finger
SPECTRAL	finger scanner, finger reader, finger sensor, fingerprint reader, iris scanner, fingerprint sensor, finger print fingerprint scanner, fingerprint reading, fingerprint recognition, same finger, **face recognition, facial recognition, have face**

Table 2: Cluster Labels

HDBSCAN	card, **battery**, sim, thing, **screen**, charge, call, purchase, back, condition, work, service, **fingerprint**, big, **finger**, love, case, upgrade, **picture**, **photo**, review, issue, problem, store, new, unlock, feature, seller, el, fast, item, one, note, well, use, mobile, small, first, network, power, size, la, cable, charger, right, certain, **price**, box, button, high, scratch, return, device, speaker, **camera**, quality, good, experience, wireless, option, model, version, **memory**, user, update, **storage**, **display**, different, carrier, mode, bottom, great
SPECTRAL	use, different, big, mode, feature, call, charge, **screen**, problem, version, scratch, device, service, work, update, size, **picture**, small, **fingerprint**, fast, review, quality, el, purchase, condition, **battery**, **storage**, return, box, mobile, new, **price**, thing, network, note, **camera**, issue, well, charger, model, card, visible, high, item, case, seller, good, first, one, carrier, great, support, original, other, speaker, wireless, user, unlock, back, many, love, experience, power, store

Table 3: Phrase Clustering Result

Feature	Sub-Feature Phrase
Screen (61)	screen display, screen size, inch display, screen resolution, screen brightness, screen sensitivity, screen ratio, lcd screen, oled screen, screen clarity, huge screen, large screen, big screen, whole screen, small screen, screen edge, curved screen, flat screen, edge screen, infinity screen, screen protector, touch screen, etc.
Memory (17)	gb memory, storage capacity, internal memory, more memory, extra memory, expandable memory, gb ram, more storage, enough space, great storage, extra storage, internal storage, storage space, additional storage, gb storage, more space, space grey
Camera (57)	front camera, selfie camera, rear camera, main camera, mp camera, camera lens, camera quality, camera app, camera function, camera software, camera upgrade, well camera, camera shutter, camera sound, camera issue, good camera, decent camera, great camera, camera noise, fantastic camera, camera work, etc.
Battery (32)	battery capacity, mah battery, battery charge, battery life, battery percentage, battery saver, battery health, battery power, battery replacement, replaceable battery, removable battery, battery drain, low battery, large battery, big battery, battery issue, battery level, battery performance, battery condition, battery quality, etc.
Security (14)	fingerprint reader, fingerprint sensor, fingerprint scanner, fingerprint reading, fingerprint recognition, finger print, finger scanner, finger reader, finger sensor, iris scanner, same finger, face recognition, facial recognition, have face
Price (37)	price range, price difference, price tag, decent price, affordable price, awesome price, perfect price, cheap price, excellent price, half price, retail price, reasonable price, same price, amazing price, price drop, sale price, nice price, fantastic price, discount price, fair price, extra money, great price, great value, pay plan, etc.

TF analysis, phrases in clusters were broke down into words and the frequency of each word was counted. The word that has the highest frequency within a cluster became the label of that cluster. For example, in Table 1, the most frequent word is 'fingerprint', so this cluster will be labeled as 'fingerprint' and it indicates that the cluster is related to the security feature. Table 2 shows the labeling result for all 72 clusters obtained in the previous section.

After labeling, important product features can be selected by product designers or experts. In this study, important features were defined as the features with high material cost and those having a great influence on product dimensions and appearance. Based on this definition, 11 clusters were selected among 72 clusters from HDBSCAN cluster-

Table 4: Noun Clustering Result

Center Word	Feature	Nouns
Surface (1125)	Screen	surface, screen, brightness, resolution, sensitivity, ratio, clarity, contrast, inch, touchscreen, space, rom, ram, selfie, megapixel, power, percentage, finger, print, sensor, budget, dollar, metal, music, volume, ear, husband, birthday, vacation, trip, pocket, step, youtube, etc.
Speed (28)	Memory	storage, memory, processor, speed, performance, display, size, camera, photo, picture, pic, video, fingerprint, face, recognition, price, value, money, feature, sound, speaker, quality, color, shape, job, design, reception, build
Battery (1)	Battery	battery
Life (3)	Battery	life, health, capacity

Table 5: Evaluation Rubric

Accuracy	Does the result contain noise words?
	Do phrases in a cluster represent the same feature?
Level	How many sub-features can be identified?

ing. Likewise, among 72 clusters from spectral clustering, 7 clusters were selected. The selected labels are marked in boldface in Table 2. Each label is assigned to one of 6 feature categories: screen, memory, camera, battery, security, and price. For the HDBSCAN clustering result, assignments are { screen: [screen, display], memory: [memory, storage], camera: [camera, picture, photo], battery: [battery], security: [fingerprint, finger], price: [price] }. For the spectral clustering result, assignments are { screen: [screen], memory: [storage], camera: [camera, picture], battery: [battery], security: [fingerprint], price: [price] }.

5 Result & Discussion

At the end of the process, the methodology provides pairs of feature topics and cue phrases. Table 3 shows the selected 6 feature topics and associated cue phrases. Some of the other clusters are presented in the appendix. In Table 3, the number in parentheses means the number of extracted phrases for each topic. The results can be evaluated in two aspects shown in Table 5: (i) accuracy; (ii) the level of extracted features.

Regarding accuracy, the previous method has downsides due to a large portion of noise included in the clustering process. Table 4 shows the noun clustering result for the smartphone review data. One distinctive feature of the result is that most data points are gathered in one cluster. This is because the noun vectors have a dense distribution in the center as shown in Fig. 5a, which forms one big cluster with X-means clustering [32]. In Table 4, the first row is that big cluster. Its center word is 'surface' and this word is related to the screen feature. However, the cluster contains many noise words such as metal, husband, birthday, vacation, and trip, which reduces the accuracy of the cluster. There is another case that compromises the accuracy of clustering results. The second row of Table 4 shows the cluster with the center word 'speed'. It represents the memory feature but words relevant to different features are mixed in this cluster. It contains 'display' which is a term for the screen feature and 'camera, photo, picture' which are related to the camera feature. Words for the security feature (fingerprint, face, and recognition) and price feature (price, value, and money) are also included in the same cluster. The first cluster also has the mixed feature problem in addition to the noise word problem. On the other hand, the proposed methodology removes a significant amount of noise before clustering, enhancing the accuracy of feature extraction. In Table 3, different feature categories are divided into separate clusters.

The second criterion is the level of extracted features. The previous method obtains limited cue phrases since it considers nouns only. The new methodology expands the range of extracted terms by considering non-noun phrases as well as noun phrases. The results in Table 3 and Table 4 show that phrase clustering extracts more specific feature terms than noun clustering. For the memory feature, the extracted terms from noun clustering are 'storage' and 'memory'. Although these words refer to the memory feature, they cannot distinguish between sub-features of memory. On the other hand, the new methodology can divide sub-features of memory by extracting phrases such as 'GB RAM', 'internal memory', and 'extra storage'. Even when comparing the total extracted feature terms aside from the clustering results, the new methodology detects more diverse sub-features than the previous method. For the camera feature, the new methodology extracts phrases for the front camera, rear camera, megapixel, and camera functions while the previous method has nouns for the front camera and megapixel. About the security feature, the new method can distinguish fingerprint, face, and iris whereas the previous method detects two of them. These results show that the proposed methodology enhances the level of extracted features so that it can give practical implications for product design.

In addition to the qualitative analysis, the results were evaluated by quantitative analysis. In the field of NLP, many studies use precision (P), recall (R), and F1 as evaluation indicators [33]. The definition is shown in Eq. 3 where tp means true positive which is predicted as positive and is actually positive. fp represents false positive that is predicted as positive but is actually negative. fn means false negative

Table 6: Result Comparison

Feature	Precision		Recall		F1		Sub-feature	
	New	Prev	New	Prev	New	Prev	New	Prev
Screen	0.967	0.020	0.855	0.100	0.908	0.038	8	8
Memory	0.941	0.071	0.727	0.333	0.821	0.118	3	0
Camera	0.807	-	0.885	0.000	0.844	-	5	0
Battery	0.875	1.000	1.000	0.500	0.933	0.667	3	2
Security	0.929	-	0.765	0.000	0.839	-	3	0
Price	0.811	-	0.811	0.000	0.811	-	0	0

* Clusters for camera, security, and price were not obtained from the previous methodology.

which is predicted as negative but is actually positive.

$$P = \frac{tp}{tp + fp} \qquad R = \frac{tp}{tp + fn}$$
$$F1 = \frac{2}{(1/P) + (1/R)} \tag{3}$$

Since Table 3 and 4 were obtained from unsupervised learning, the result was evaluated by a manual process. All items in the tables were classified as one or more of three categories (tp, fp, fn). The category was determined by a domain expert with 10 years of work experience in the smartphone industry. For example, in the 'Memory' cluster of Table 4, 'storage' is a correct keyword for the memory feature, so the expert classified it as tp. The cluster contains 'fingerprint" which is related to the security feature. Therefore, the expert classified 'fingerprint' as fp. 'ram' is a keyword for the memory feature, but it belongs to the 'Screen' cluster. In other words, 'ram' is predicted to be unrelated to memory when it is actually related to the memory feature. Therefore, the expert classified 'ram' as fn. Based on this classification, three evaluation indicators were computed. Precision calculates the ratio of correct keywords to all keywords within a cluster. Recall computes the proportion of correctly detected keywords among all keywords related to a feature. F1 measures the harmonic mean of precision and recall. Table 6 compares these three indicators of the two methodologies. In the previous method, precision and F1 cannot be computed for some features because the clusters corresponding to those features were not obtained. Aside from that, for almost all features, the new methodology provides higher precision and recall rates. Regarding F1 measure, the new method outperforms the previous one for all features. The diversity in the result was evaluated by counting the number of detected sub-features. Table 6 shows that phrase clustering extracts more sub-features than noun clustering. This evaluation shows that the proposed method improves the performance of feature extraction in terms of accuracy and diversity.

6 Conclusion & Future Works

This paper focuses on the gap between research and industry in the product design area. As mentioned in Section 1, a product is manufactured by configuring multiple components. However, studies about customer analysis using online data have been focusing on feature categories rather than product components. The implications for feature categories may help product designers set a design strategy or direction, but they are not specific enough to be used in embodiment design. To address this limitation, this paper proposes a new methodology that extracts component-level features from online user-generated data. First, the proposed methodology collects phrases in online data and models them into a vector space. Then, these phrase vectors are grouped into several clusters by two clustering methods. The resulting clusters are labeled by TF analysis of items in each cluster. Finally, by selecting cluster labels of interest, product designers can obtain sub-feature phrases mentioned by online customers.

The suggested methodology was tested on smartphone reviews and compared with the previous methodology. The qualitative evaluation shows that the proposed method addresses shortcomings of the previous one. The new method removes noise words from the feature clusters, and it divides different feature categories into separate clusters while the previous method merged some features into one cluster. Also, the new method enhances the diversity of extracted sub-features. In the quantitative analysis, evaluation indicators including precision, recall, and F1 score show the improved performance of the proposed methodology. Regarding phrase embedding, there is another approach that can be considered for this topic. Wu et al. [34] train phrase vectors using Skip-Gram in Fig. 1b, the most classic Word2vec model. The authors evaluate the performance of Phrase2vec, but the application of the method is not presented in their paper. The Phrase2vec model can be tested for sub-feature extraction in the extended work of this study.

For future works, we will conduct customer analysis to draw implications for embodiment design. First, a keyword dictionary for sub-features will be created based on the result of this study. The dictionary will define cue phrases for each sub-feature. An example is {Screen size: [screen size, inch, large screen, small screen], Screen technology: [screen technology, lcd screen, oled screen], Security fingerprint: [fingerprint reader, fingerprint sensor, finger scanner], Security face: [face recognition, facial recognition], Security iris: [iris scanner]}. The online customer reviews will be analyzed

Journal of Mechanical Design. Received April 26, 2021;
Accepted manuscript posted October 15, 2021. doi:10.1115/1.4052904

based on the dictionary. This will detect the reviews containing target sub-features. Once the reviews are detected, customer preferences for the sub-feature can be analyzed. Chaklader & Parkinson [35] proposed a method using customer ratings. They draw proper spec ranges by comparing the overall rating to the average rating of reviews mentioning the target feature. Sentiment analysis can also be used for extracting customer preferences [14, 15].

The customer analysis based on sub-features will give design implications for embodiment design. It can analyze the importance of each sub-feature or customers' preferences for spec ranges. This will help companies make better decisions about product configuration by providing component priorities. Also, it can help product designers find a better design by adding new constraints - recommended spec ranges - in a design optimization problem.

References

[1] Pahl, G., Beitz, W., Feldhusen, J., and Grote, K. H., 2007. *Engineering Design*. Springer.

[2] Sudin, M. N., and Ahmed, S., 2009. "Investigation of change in specifications during a product's lifecycle". In Proceedings of ICED 09, the 17th International Conference on Engineering Design, pp. 371–380.

[3] Chevalier, J. A., and Mayzlin, D., 2006. "The effect of word of mouth on sales: Online book reviews". *Journal of Marketing Research,* ***43***(3), Aug, pp. 345–354.

[4] Sun, M., 2012. "How does the variance of product ratings matter?". *Management Science,* ***58***(4), Apr, pp. 696–707.

[5] 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". *International Journal of Production Research,* ***55***(17), pp. 5142–5156.

[6] Hu, M., and Liu, B., 2004. "Mining opinion features in customer reviews". *19th National Conference on Artifical Intelligence, San Jose, CA,*, Jul, pp. 755–760.

[7] Manning, C. D., Raghavan, P., and Schütze, H., 2009. *An Introduction to Information Retrieval*. Cambridge University Press.

[8] Abulaish, M., Jahiruddin, Doja, M. N., and Ahmad, T., 2009. *Feature and Opinion Mining for Customer Review Summarization*. Springer.

[9] Blei, D. M., Ng, A. Y., and Jordan, M. I., 2003. "Latent dirichlet allocation". *Journal of machine Learning research*, Jan.

[10] Mei, Q., Ling, X., Wondra, M., Su, H., and Zhai, C., 2007. "Topic sentiment mixture: modeling facets and opinions in weblogs". *Proceedings of the 16th international conference on World Wide Web, WWW '07*, May, pp. 171–180.

[11] 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". *Journal of Electronic Commerce Research,* ***14***(4), pp. 304–314.

[12] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J., 2013. "Distributed representations of words and phrases and their compositionality". In Advances in neural information processing systems, pp. 3111–3119.

[13] Joung, J., and Kim, H. M., 2020. "Automated keyword filtering in lda for identifying product attributes from online reviews". *J. Mech. Des.*

[14] Tuarob, S., and Tucker, C., 2015. "Quantifying product favorability and extracting notable product features using large scale social media data". *Journal of Computing and Information Science in Engineering,* ***15***(3).

[15] Suryadi, D., and Kim, H., 2018. "A systematic methodology based on word embedding for identifying the relation between online customer reviews and sales rank". *Journal of Mechanical Design,* ***140***(12).

[16] Sapuan, S. M., 2017. *Composite Materials*. Butterworth-Heinemann.

[17] Mikolov, T., Chen, K., Corrado, G., and Dean, J., 2013. "Efficient estimation of word representations in vector space". *arXiv preprint arXiv:1301.3781*.

[18] Rong, X., 2014. "word2vec parameter learning explained". *arXiv preprint arXiv:1411.2738*.

[19] Jain, A. K., 2010. "Data clustering: 50 years beyond k-means". *Pattern Recognition Letters,* ***31***(8), Jun, pp. 651–666.

[20] Xu, D., and Tian, Y., 2015. "A comprehensive survey of clustering algorithms". *Annals of Data Science,* ***2***(2), pp. 165–193.

[21] Har-Peled, S., 2011. *Geometric Approximation Algorithms*. No. 173. American Mathematical Soc.

[22] Lloyd, S. P., 1982. "Least squares quantization in pcm". *IEEE TRANSACTIONS ON INFORMATION THEORY,* ***28***(2), Mar, pp. 129–137.

[23] Luxburg, U., 2007. "A tutorial on spectral clustering". *Statistics and computing,* ***17***(4), pp. 395–416.

[24] Cortesy, A., Chaki, N., Saeed, K., and Wierzchon, S., 2012. *Computer Information Systems and Industrial Management*. Springer.

[25] Campello, R., Moulavi, D., and J., S., 2013. *Density-Based Clustering Based on Hierarchical Density Estimates*. Springer.

[26] Zhang, D., Xu, H., Su, Z., and Xu, Y., 2015. "Chinese comments sentiment classification based on word2vec and svmperf". *Expert Systems with Applications,* ***42***(4), pp. 1857–1863.

[27] Pelleg, D., and Moore, A. W., 2000. "X-means: Extending k-means with efficient estimation of the number of clusters". In ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 723–734.

[28] Zhang, L., Li, J., and Wang, C., 2017. "Automatic synonym extraction using word2vec and spectral clustering". *Proceedings of the 36th Chinese Control Conference*.

[29] O'Dea, S., 2021. "Smartphones in the u.s. - statistics & facts". `https://www.statista.com/topics/2711/us-smartphone-market/`.

[30] Wold, S., Esbensen, K., and Geladi, P., 1987. "Principal component analysis". *Chemometrics and Intelligent Laboratory Systems*(1-3), pp. 37–52.

[31] Halkidi, M., Batistakis, Y., and Vazirgiannis, M., 2001. "On clustering validation techniques". *Journal of intelligent information systems,* ***17***(2), pp. 107–145.

[32] Park, S., and Kim, H. M., 2020. "Improving the accuracy and diversity of feature extraction from online reviews using keyword embedding and two clustering method". In Proceeding of ASME 2020 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.

[33] Lipton, Z., Elkan, C., and Naryanaswamy, B., 2014. *Optimal Thresholding of Classifiers to Maximize F1 Measure*. Springer.

[34] Wu, Y., Zhao, S., and Li, W., 2020. "Phrase2vec: phrase embedding based on parsing". *Information Sciences,* ***517***, pp. 100–127.

[35] Chaklader, R., and Parkinson, M. B., 2017. "Data-driven sizing specification utilizing consumer text reviews". *Journal of Mechanical Design,* ***139***(11).

A Appendix: Phrase Clustering Result

Among 72 clusters, 6 clusters are presented in Table 3. From the remaining 66 clusters, 10 clusters are selected and presented. They include both feature-related clusters and non-feature-related clusters. The first column shows the cluster label which is automatically determined by the proposed methodology.The second column shows the phrases in the cluster.

Table 7: Phrase Clustering Result

Label	Phrase
Charge	charge port, full charge, charge cable, charge cord, charge plug, quick charge, wireless charge, etc.
Call	voice call, wifi call, video call, miss call, drop call, incoming call, call volume
Cable	lightning cable, generic cable, usb cable, usb cord, charging cable, c cable, c port, usb adopter, etc.
Speaker	loud speaker, ear speaker, stereo speaker, internal speaker, external speaker
Button	home button, power button, volume button, side button, bixby button
Network	cdma network, gsm network, wireless network, cellular network, mobile network, etc.
Service	customer service, call service, service provider, verizon service, att service, sprint service, etc.
Box	amazon box, verizon box, white box, generic box, seal box, plain box, cardboard box, etc.
Scratch	visible scratch, physical damage, minor scratch, scratch crack, scratch dent, noticeable scratch, etc.
Return	return process, return label, return shipping, return request, return date, return policy, etc.