CIRP Journal of Manufacturing Science and Technology 23 (2018) 128–144

Contents lists available at ScienceDirect

CIRP Journal of Manufacturing Science and Technology

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

Application of data analytics for product design: Sentiment analysis of online product reviews

Robert Ireland, Ang Liu*

School of Mechanical and Manufacturing Engineering, University of New South Wales, Sydney, NSW 2052, Australia

ARTICLE INFO

Article history:
Available online 17 July 2018

Keywords:
Design method
Sentiment analysis
Data analytics
Online product review

ABSTRACT

Advanced data analytics is one of the most revolutionary technological developments in the 21st century, which enables the discovery of underlining trends via sophisticated computational methods On various e-commerce and social platforms, millions of online product reviews are published by customers, which can potentially provide designers with invaluable insights into product design. This paper presents a design framework to analyze online product reviews. The objective is to use this machine-generated data to identify a series of customer needs. The framework aims to distill large volumes of qualitative data into quantitative insights on product features, so that designers can make more informed decisions. The framework combines the elements of online product reviews, design theory and methodology, and data analytics to reveal new insights. The effectiveness of the proposal framework is validated through a case study on product reviews from the e-commerce website, Amazon. The framework demonstrates a statistical approach for analyzing online product reviews. The framework acts as an interface between quantitative outputs and the qualitative and creative process of design. Further analysis of results identifies many of incorporating logical, computational methods into the highly subjective and creative process of design.

Introduction

In light of the sweeping trend of data analytics, a lucrative area of research involves how the vast amount of customer-generated data online can be used by designers to develop more competitive products [1]. The enormity and rapidity by which online product reviews (OPRs) are generated on e-commerce websites, social media and blogs offer a wealth of information for manufacturers to improve product design. It has been suggested by many previous studies that OPRs contain valuable information for the early stages product design [2–7]. However, OPRs consist of variables that are difficult to analyze by computation. A machine can effectively analyse quantitative feedbacks, however, understanding the tone, feelings, irony, and context of human sentiment presents variables that are far more difficult to quantify.

The importance of soliciting customer voices and understanding their need and wants has long been highlighted by design theories and methodologies. According to Axiomatic Design [8], any design process begins with a set of customer needs (CNs), or interpreting what customers desire in a product. They are then mapped to functional requirements (FRs), which set the product performance standards necessary to meet the CNs. Traditionally, CNs are manually solicited by means of surveys, interviews, focus groups, ethnographic observation, lead user theory, etc. The Kano Model employs customer satisfaction to categorize product features. Typically, the implementation of the Kano Model involves surveying customers about their sentiments towards product features [9]. The information is then used to generate CNs, which are then correlated in a matrix form to create FRs [9]. Quality Function Deployment (QFD) involves translating the “Voice of the Customer” into product requirements [10]. In practice, both QFD and Kano Model are proven highly useful for indicating the relative importance of different product features in correspondence to CNs. However, they both require significant expertise from a designer and usually extract information from a small group of customers. Hence, they are often expensive and time-consuming to use. There are new methods developed for incorporating data analysis into QFD and Kano Model [11–15], although the final results still require interpretation by a human designer.

Sentiment analysis refers to the computational process of recognizing and classifying various opinions expressed in text. The purpose of sentiment analysis, often called “opinion mining”, is to

* Corresponding author.
E-mail address: ang.liu@unsw.edu.au (A. Liu).

https://doi.org/10.1016/j.cirpj.2018.06.003
1755-5817/© 2018 CIRP.

determine the writer's attitude (positive, negative, or neutral) towards a particular item [16]. Despite the rapid advancements of sentiment analysis in other fields, the links between sentiment analysis and product design remain relatively unexplored.

This paper presents a design sentiment analysis framework. The input of the framework is a selection of OPRs published by customers on the e-commerce platform (e.g., Amazon.com), whereas the output is a set of categorized customer opinions towards a product. The framework is characterized by an integration of key natural language processing (NLP) techniques and machine learning algorithms. Most importantly, a structured computational process, known as the Machine Model, is prescribed to automatically perform sentiment analysis on given OPRs. A case study is presented to showcase effectiveness of the framework. Specifically, the Machine Model is compared to a Human Model, with human-generated output, in terms of their design performance of analyzing the same set of product reviews. The Human Model serves as the control to evaluate the effectiveness of the Machine Model. The two sets of results showcase how designers can use the Machine Model to generate CNs and assess its accuracy relative to sophisticated human cognition.

Literature review

Sentiment analysis of unstructured data

Online product reviews (OPRs) come in an unstructured form [17]. Different from structured data that are composed of easily quantifiable information such as website views or viewer locations, unstructured data are overwhelming in terms of both quantity and variety [18]. The complexity of human expression presents a fundamental challenge in the computational interpretation of OPRs [19]. Most of the existing methods for processing unstructured data involve identifying individual words, while few can understand the meaning of full sentences or paragraphs [20]. Though it is fairly simple for human designers to understand unstructured data, the large number of variables and assumed knowledge a human possesses is challenging for machines to replicate.

The area of NLP most relevant to this research is the sentiment of sentences that describe product features. A positive or negative sentiment towards a product feature indicates to the designer the specific CNs they must enhance. By definition, sentiment means an attitude, thought or judgment prompted by a feeling [21]. The "semantic orientation" towards a product feature pertains to whether it is liked or disliked by the user [16]. For example, consider the phrases "I hate the chair" versus "I like the chair." Understanding that the connotation of "like" is positive and of "hate" is negative conveys its semantic orientation. Fang and Zhan noted that the negation is also important to consider; for example, when analyzing the phrases "I like the chair" and "I do not like the chair," it is imperative that the "not" reverses the sentiment to negative [21].

A study by Yin et al. detailed several NLP algorithms to extract the sentiment from OPRs [2]. By tracking how sentiments changed over time as new features were added, designers were enabled to quantitatively assess the impact of continuous design improvements. Liu et al. found that the context of how a word was used was vital to infer opinions from sentences [22], by comparing the sentiment towards features of competing products, so designers can identify the most competitive product with the most favorable features [22].

One example of new linguistic methods used to infer sentiment is Ding et al.'s "intra-sentence conjugation rule" [16]. Using this rule, the sentiment of "the battery lasts long" could be inferred from a positive conjugation of another word such as "the camera is great and the battery lasts long [16]." Thus, the use of the adjective "long" is deemed to be of positive sentiment. Lee's "subject-verb-object triple" model also aimed to find co-occurrences of sentiments that occur with features [23,24]. This allows for an easily interpretable set of results indicating the importance of features.

Similarly, Hu and Liu follow the "feature-based opinion summarisation" process to discover trends in product features [6]. They employed two linguistic techniques to identify the syntactic description of the terms. The first was to identify the noun and the second to statistically model how close the adjective is found to the noun, followed by how often it occurs [6]. Hu and Liu also highlighted the difficulty in analyzing implicit feature sentiments compared with explicit features [6]. The example given of an implicit sentiment of a feature is that a camera "will not fit easily in a pocket" implies that the size is too big. However, a computer cannot make this inference. An explicit sentiment is far easier to qualify, such as "the camera is too big."

Using natural language processing to analyse online product reviews

In addition to sentiment analysis, there are recent developments in how designers can benefit from OPRs. Designers require a far more sophisticated amount of intelligence than whether the product feature is liked or disliked. They must go beyond and understand the entire user experience of the product [7]. Though existing models may identify the sentiment of a specific word and map it with a feature, understanding the context from which problems arise can be difficult to extract [25]. In fact, it is commonly acknowledged that most online opinion mining is not effectively utilized by designers [4].

Specific use cases of products can be as important to the designer as the sentiment towards its features. Sentiment analysis can identify if customers are unsatisfied with a feature, however, it cannot infer why this came about. For example, an NLP module can conclude customers are unsatisfied with a phone's fragile screen, though they may miss key information about how customers resolve the problem. Jian et al. aimed to solve this dilemma by identifying product features and then collating associated verbs and adjectives based on their sentiment [5]. As a result, a designer is presented with a list of associated positive and negative words with each feature. Designers can then review this list and make their own assessments. Despite the ease of interpreting results in binary categories, when a sentiment is taken out of context, its usefulness to designers becomes limited.

Despite advances in sentiment analysis, sentiment does not provide designers with the detailed context of what causes the sentiment. For example, a review may convey negative sentiment towards a "battery" feature, but not define what problem the battery causes [5]. To extract context from OPRs, Jian et al. developed a framework to identify four elements of reviews: product features (F), sentiment polarity (S), aspects of features (A) and detailed reasons (R) [5]. Though the model provided the designer with greater information of the reasons behind customer sentiment, the extraction of aspects and reasons out of context is difficult to interpret. If results are presented out of context, it is best for them to be organized the most interpretable manner such as a graph or diagram. Otherwise, designers may not find them helpful.

To solve the issue of information loss as data is distilled, Jin et al. proposed a model where sentences were evaluated on their quality and relevance to key features [4]. The model returned the best quality sentences for the designer to review manually [4]. Using a small amount of high-quality intelligence was faster than analyzing a high volume of reviews, yet provided them with the data to make informed design decisions. The model does not, however, draw attention to reviews of lower quality, even if they are numerous and highlight a key issue with a feature. Its qualitative nature allows the designer to understand the context but not the statistical significance of product trends.

Another method used to extract designer intelligence involved using associative algorithms to identify common phrases. Chung and Tseng constructed two mathematical methods to determine relationships between important features; one using the associative Apriori algorithm and the other using rough set theory [26]. To identify key words, they employed the term-frequency inverse document frequency (TF-IDF) algorithm to determine what words were important, and then other Apriori and other algorithms to uncover patterns associated with these words [26]. Using a purely mathematical approach to uncover associations makes it easy to replicate and process large datasets, however, it removes descriptive details.

When using NLP for the design process, a dilemma always exists between informative qualitative data and developing statistically significant quantitative data. The translation of qualitative data to quantitative can oftentimes hide the “why” and “how” behind customer sentiments. Conversely, providing the reasoning behind these “why” and “how” issues is best achieved in language and not by statistical means. For example, statistical methods can easily identify whether customers generally like or dislike product. Though to communicate the reasons why cannot be done so easily by numbers; they require a more comprehensive and linguistic explanation. Achieving a balance between these mutual exclusivities is an essential component for the incorporation of NLP into design.

Integration between design theory methods with natural language processing

The challenge between quantitative and qualitative data is an established issue within design research itself. Nonetheless, some models address the shortcomings of each model type by implementing both a traditional qualitative design process and quantitative data analysis.

One such method is to gather intelligence on specific features and then employ a manual design methodology to interpret the data and make informed decisions. Jin et al. developed a computational module to determine what were the most important features of a product, and then evaluated them by Quality Function Deployment (QFD) to rank their importance [3]. By limiting the scope and allowing designers to formulate CNs based on data, many of the issues that occur when understanding the context behind a sentiment were eliminated. A key problem with generating CNs from OPRs is that many important functions are often overlooked by customers [3]. Therefore, designers must regard machine-generated knowledge with circumspection, as it is not yet a complete substitute for human knowledge and wisdom.

A critical factor for consideration when using OPRs as data in design is how accurately they determine CNs. One study by Jin et al. aimed to compare how accurately online product reviews would address key engineering characteristics of a product [27]. A QFD process was completed by designers to establish key engineering characteristics, followed by a Bayesian probability model, which ascertained whether reviews addressed important engineering characteristics [27]. It revealed the difficulty of abstracting functional characteristics of products from OPRs. In other words, OPRs are useful for the extrapolation of CNs, not Functional Requirements (FRs).

Qi et al. involved mining OPRs to develop customer requirements by using a Kano design methodology [28]. The Kano method uses a set of parameters to determine the customer satisfaction of a product. These include requirements essential to the product’s basic function and those that are non-essential to function but attractive to customers [28]. Typically, Kano customer satisfaction is measured by surveys, though this is not easily transferable into an NLP platform. By using mathematical methods such as conjoint analysis and the Support Vector Machine (SVM), it was possible to categorize features according to the Kano Model [28].

It is clear that existing design methodologies such as QFD and Kano are difficult to apply with NLP. Therefore, the proposed framework aims to focus on generating the most effective and interpretable information for the development of CNs. It is presumed that machine-learning and NLP methods are most advantageous in the earliest stages of design, as machines can simplify large amounts of data but not make decisions for product design.

Sentiment analysis design framework

This section presents the design sentiment analysis framework with respect to the input data of online product reviews (OPRs) (see Section “Online product reviews as data source”), the relevant natural language processing (NLP) techniques (see Section “Natural language processing techniques”) and machine learning algorithms (see Section “Machine learning algorithms”), as well as a machine sentiment analysis process (see Section “Machine Model of sentiment analysis”).

Online product reviews as data source

Amazon.com was selected as the data source of online product reviews (OPRs). As one of the largest e-commerce platforms in the world, Amazon.com offers a broad range of products and millions of OPRs available for analysis. By demonstrating the developed framework’s applicability on a single product from Amazon, it can be applied to millions of other products. Reviews published on Amazon.com contain useful information such as reviewer identification, reviewer credibility, product rating, time of review, helpfulness as judged by other reviewers, and the ability to edit comments at a later date [31]. In addition to the large quantity of data, Amazon’s culture of encouraging customer feedbacks ensures that it is a high-quality data source. This is exemplified by the “Hall of Fame” on Amazon.com — a page glorifying their most helpful customers [29].

The quality of reviews must be rigorously evaluated. A simple measure of quality on Amazon reviews is the “helpfulness” of reviews as voted by other peer customers. A study by Liu et al. showed that the “helpfulness” rating on Amazon did not always correlate with the helpfulness rating determined by designers [30]. Liu et al. defined four key features to evaluate helpfulness: linguistic, product, information quality, and information theory [30]. Linguistic features include elements such as number of descriptive words and grammar, product features include important feature of the product itself, information quality features pertain to sentiment and subjectivity and information theory features relate to the divergence of sentiment in the review [30].

Natural language processing techniques

Since product reviews consist of text written by customers, multiple key NLP techniques are incorporated into the framework. NLP bridges the study of linguistics and how human language is used with various statistical methods.

WordNet

WordNet is an online lexical database for interpreting the definition of a word and how similar it is to another [31]. WordNet refers to these related words as “synsets”, which consist of related groups of nouns, adjectives, verbs, and adverbs [32]. These synsets can be represented in a tree-like structure as shown in Fig. 1 [32].

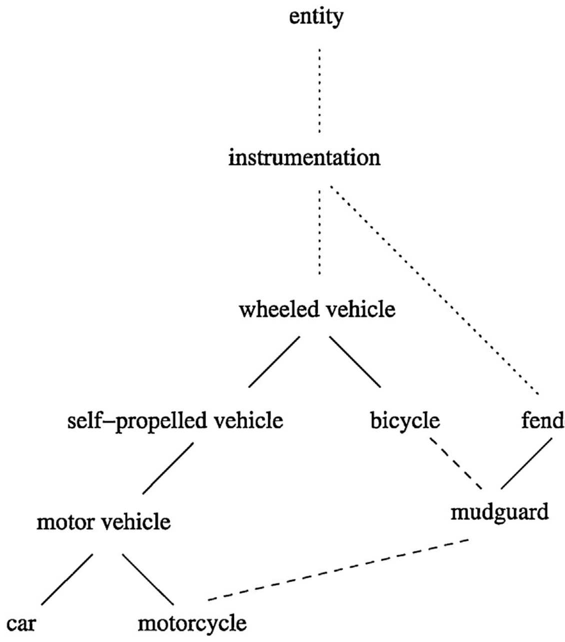


Fig. 1. Example of the relationship between words in WordNet by Scriver [42].

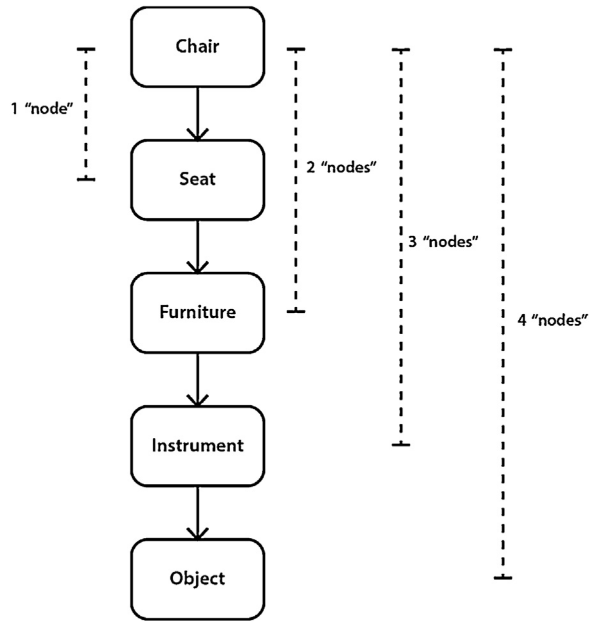


Fig. 2. WordNet hierarchy of synsets.

WordNet synsets involve degrees of relation according to methods derived from linguistics. WordNet can identify synonyms, which are words of similar meaning, and the least related are antonyms [31]. It can also differentiate hyponyms and hypernyms. Hyponyms refer to a specific type of a general class; for example, a "robin" is a hyponym of "bird", and "bird" a hypernym of "robin" [32]. Similarly, meronyms and holonyms are also measured, where meronyms are parts of a whole and holonyms the whole of the parts [31]. For example, a "piston" is a meronym of an "engine", and an "engine" a holonym of a "piston." In addition, there are advanced functions of WordNet that can identify malapropisms, or associate misspelled words with their equivalent words [32].

The similarity or relatedness between any two words is often referred to as their "WordNet distance." Despite the complexity and variation of word meanings, there are some processes used to identify degrees of relatedness. One such model is determining "nodes" between synsets [32]. In this model, "nodes" represent one level of relatedness between a synset. For example, a synonym is of one node away, and hypernym/hyponyms and meronyms/holonyms represent two nodes. Fig. 2 shows the level of association between synsets that constitute a node.

The interpretation of words is universally qualitative, subjective and complex. WordNet itself was constructed by mapping the relationships humans associate with words, which is inherently subjective. One key limitation of WordNet is that it cannot always identify the context from which a word is taken [33]. To measure this subjectivity, Boyd-Graber et al. surveyed a number of people on how words "evoke" a relationship with another to provide insight into the context in which words are used [33]. It should be noted that WordNet is constantly evolving to adapt to the subjectivity of language, epitomizing the challenge of using statistical methods for language analysis.

Part-of-Speech Tagger

The Part-of-Speech (POS) Tagger is commonly used to identify different types of words in NLP. There are many POS Tagger programs available, most of which stem from the Stanford Maximum Entropy POS Tagger [34]. The Stanford POS Tagger is generally very accurate, using WordNet as a database and Maximum Entropy to predict the correct word type (see Section "Step (1): tag word types via Part-of-Speech (POS) Tagger"). This is particularly useful to differentiate polysemous words or the same words with different meanings [35].

PlingStemmer

PlingStemmer is one of the many tools that can strip words down to their most basic singular form, making it easier for other programs to process sentences [36]. The singular form of words, known as their "lemma", makes it easier for NLP programs to identify words and make associations. For example, PlingStemmer will identify the words "broke", "broken" and "breaking" simply by the singular present form "break." However, there are many exceptions to determining if a suffix determines whether a word is singular or plural [36].

Cosine similarity measure

The cosine similarity measure (CSM) is an algorithm that determines how similar different textual documents are to each other. The algorithm works by assigning a vector to each word and then measuring the angular difference between words on a large scale. The algorithm can be shown below in Eq. (1), with Table 1 outlining each symbol [37].

$$\frac{\cos(\theta) = \vec{a}\cdot\vec{b}}{\|a\|\|b\|} \tag{1}$$

Since similarity is calculated based on the angle of the vectors, the document size becomes less important as it only increases the Euclidean distance [38]. CSM is very accurate at determining document similarity regardless of documents size. It can determine the similarity of input documents by calculation of an angle [39]. The more similar the output angles of documents, the more alike they are. In NLP, it can be used to find reviews or even

Table 1
Cosine similarity measure equation values.

Value	Meaning
$\cos\theta$	Angular difference between vectors a and b
$\vec{a}$	Vector of document a
$\vec{b}$	Vector of document b

sentences that relate to a specified topic. For example, if certain features were considered more important, CSM would be an efficient way to find reviews that mention these certain features.

Machine learning algorithms

Since the machine is expected to actively learn from the previous analysis for more accurate predictions, multiple key machine learning algorithms are incorporated into the framework, including Term Frequency-Inverse Document Frequency (TF-IDF), Naïve Bayes, Maximum Entropy, Support Vector Machine and Apriori.

Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is an effective machine learning algorithm to determine the relative importance of words in a document [40]. It weights a word more heavily the more often it occurs in a single document and inversely by how often it occurs in a large selection of documents. As a result, words such as “the”, “as” and “it” are deemed less important compared with other nouns, adjectives and verbs. The TF-IDF equation, Eq. (2), is split into two components shown by Eqs. (3) and (4). Term frequency (TF) assigns a weight for how often the word occurs in a single review. Inverse document frequency (IDF) determines the word’s importance by counting how often the word appears in other reviews [41]. Table 2 shows each of the values used in the equations.

$$TFIDF = TF \times IDF \tag{2}$$

$$TF = \frac{f_i}{f_r} \tag{3}$$

$$IDF = \log_e\left(\frac{N}{N_i}\right) \tag{4}$$

TF-IDF can sometimes disregard the importance of similar words [42,43], which in this research is a significant problem when most reviews refer to the same features. Its application becomes more relevant as the data size increases. When analyzing thousands of reviews, it may be an efficient way identifying key words or phrases. For example, it might be too computationally expensive to analyze all product features in sentences when there are over 10,000 reviews. TF-IDF would immediately be able to identify the most relevant sentences to streamline the analysis. One study used TF-IDF to determine which words in a document were the most important, and then used CSM to find the most relevant documents [37].

Naïve Bayes classifier

One of the most effective machine learning algorithms to predict whether a word is positive, negative or associated with another word is the Naïve Bayes (NB) algorithm. NB acts as a classifier to determine if a word is more likely to be positive or negative in sentiment. The probability that a sentence is positive or negative is the product sum of the probability of whether each word in that review is positive or negative is shown by Eq. (5). Table 3 below describes the notations used in the following NB Equations.

Table 2
TF-IDF equation values.

Value	Meaning
f_i	Number of times a term “i” appears in a review
f_r	Total number of terms in each review
N_i	Number of reviews containing term “i”
N	Total number of reviews

$$P(review|v_j) = \prod_{i=1}^{length\ of\ review} P(a_i = w_k|v_j) \tag{5}$$

The process to determine whether a word is positive or negative is iterative, as the probability adapts as more words are considered. If certain words are already classified as positive or negative, the probability of calculating whether a word is positive is given by Eq. (6).

$$p(w_k|v_j) = \frac{n_k + 1}{n + N_v} \tag{6}$$

After calculating the probability of whether a word is positive and negative, whichever value is higher determines the sentiment value as shown in Eq. (7).

$$V_{NB} = \mathrm{argmax}\, P(v_j)\prod P(a_i = w_k|v_j) \tag{7}$$

Unlike many other machine learning algorithms, NB does not require large training datasets or even large datasets to produce accurate results [44]. NB was selected to analyze sentiment due to its high accuracy in smaller datasets than other algorithms.

Maximum Entropy

Maximum Entropy (ME) is a classification method that involves a process of logistic regression or creation of a function to best correlate data [45]. In contrast with NB, ME measures dependent variables in theory is superior for identifying patterns [47]. If implemented it must be trained with a much larger quantity of prior data than NB otherwise it may not be accurate [48]. Once trained, ME can identify the optimal data to extract.

Regression analysis in machine learning generally involves the formation of a hypothesis function, $h_\theta(x)$, to describe the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$ planes, as shown by Eq. (8) [45].

$$\begin{bmatrix} x^1 \\ x^2 \\ x^3 \\ .. \\ x^n \end{bmatrix} \rightarrow h_\theta(x) \rightarrow \begin{bmatrix} y^1 \\ y^2 \\ y^3 \\ .. \\ y^n \end{bmatrix} \tag{8}$$

The complexity of the function can depend on whether a linear or non-linear solution is obtainable. In basic terms, the ME algorithm learns to continually optimize itself and categorize words into many classes. The formula for conditional ME is shown by Eqs. (9)–(11) [48]. Values are described in Table 4.

$$P(c|d) = \frac{1}{Z(d)}\exp\left(\sum_i \lambda_i f_i(c,d)\right) \tag{9}$$

$$Z(d) = \sum_c \exp\left(\sum_i \lambda_i f_i(c,d)\right) \tag{10}$$

$$f_i(c,d) = \begin{cases} 1\ \textit{if d is contained in document and c is the label} \\ 0\ \textit{otherwise} \end{cases} \tag{11}$$

ME is excellent for data categorization. It is incorporated into the POS Tagger to identify word types. NB like ME can be used to determine sentiment types. It is also possible for ME to categorize sentiment words with features in reviews, producing a more thorough analysis. The drawback, however, is the extensive training the algorithm requires.

Table 3
Values used in Naive Bayes equations.

Value	Meaning
$P(a_i = w_k \mid v_j)$	Probability of word a_i in sentence being a known positive or negative word
$p(w_k \mid v_j)$	Probability of a word being positive or negative
$P(v_j)$	Probability of a document being positive or negative
V_{NB}	Naïve Bayes value, either positive or negative
n_k	Number of times the word w_k occurs in positive or negative cases
n	Number of words in positive or negative case
N_v	Size of vocabulary
w_k	A known word in the vocabulary
a_i	A particular word in a document before being identified in vocabulary
v_j	Positive or negative class

Table 4
Values for Maximum Entropy equations.

Value	Meaning
$P(c \mid d)$	Probability a word 'd' is of class 'c'
c	Class or category
d	Word in a document
λ_i	Weight of importance
$Z(d)$	Normalising factor to improve accuracy of probability
$f_i(c, d)$	Feature function to determine class

Apriori algorithm

Apriori algorithm is a frequency association algorithm, which determines if certain values are related to or frequently associated with others [49]. Apriori is commonly used in online product recommendation systems to discern a customer's previous purchase patterns [50].

The Apriori algorithm uses three equations to establish association rules [50]. The first, known as Support Count, involves how frequently certain sets and sub-sets occur, with the more frequent sub-set relationship the more important the rule. The second, known as Confidence, measures the likelihood of the Support Count frequency. The third, known as Lift, measures the accuracy of the association. Apriori is governed by a principle that if an item set is frequent then all its subsets are also presumed frequent [49]. Given a set of transactions shown in Table 5, the Support Count for subset {A} is calculated by Eq. (12) below.

$$sc\{A\} = \frac{No.\ of\ transactions\ with\{A\}}{Total\ no.\ of\ transactions} = \frac{4}{5} \tag{12}$$

Thus, {A} occurs in 4 out of 5 transactions. The Confidence can then be used to determine how likely it is that {A} occurs with another subset. Below the Confidence of how often {A} occurs when {B} occurs is calculated by Eq. (13).

$$C\{A|B\} = \frac{sc\{A,B\}}{sc\{A\}} = \frac{3/5}{4/5} \tag{13}$$

The$L\{A|B\} = \begin{cases} = 1\ if\ A\ and\ B\ are\ independent \\ > 1\ if\ A\ and\ B\ are\ positively\ correlated \\ < 1\ if\ A\ and\ B\ are\ negatively\ correlated \end{cases}$ refore, it is 75% likely that a person will buy {B} if they buy {A}. However, the Confidence can be inflated using this method as it only accounts for how popular {A} is, and not {B}. A measure called "Lift" is used to account for this discrepancy as shown in Eq. (14) below.

Table 5
Transactions and corresponding purchased items.

Transactions	Purchases
1	{A} {B} {C} {D}
2	{A} {B} {C}
3	{A} {B}
4	{E} {D} {C}
5	{A} {D} {E}

$$L\{A|B\} = \frac{sc\{A,B\}}{sc\{A\} \times sc\{B\}} = \frac{3/5}{4/5 \times 3/5} = 1.25 \tag{14}$$

Generally, the following weighting associated with the Lift value is shown by Eq. (15) [50].

$$L\{A|B\} = \begin{cases} = 1\ if\ A\ and\ B\ are\ independent \\ > 1\ if\ A\ and\ B\ are\ positively\ correlated \\ < 1\ if\ A\ and\ B\ are\ negatively\ correlated \end{cases} \tag{15}$$

As $L\{A|B\} = 1.25 > 1$, it is likely that {A} will be bought if {B} is bought. The above example has only a few itemsets, however th,e combinations of subsets exponentially increases as the number of itemsets increases. Therefore, for an example with hundreds of transactions and products a process known as pruning occurs [49]. This involves setting a minimal value for Support Count, Confidence and Lift whereby low values are removed from further calculations to ensure only the most importance rules are established.

In the context of analyzing OPRs, Apriori has been used to identify associations between product features and sentiment words [51]. Apriori is excellent at making association between word pairs and other factors like customer buying habits and related products.

Support Vector Machine (SVM)

SVM is another popular algorithm used for classification [52]. Each data element is plotted in an n-dimensional space, assigning the element a vector coordinate [53]. Then SVM algorithm forms what is known as a hyperplane to differentiate one class of data from another. As Fig. 3 demonstrates, there are many variants of classifier lines (shown in green), but the optimal hyperplane

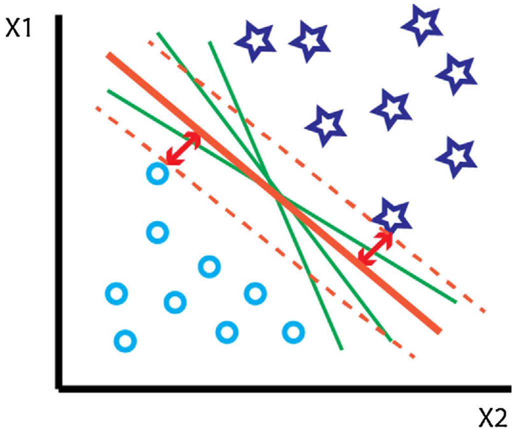


Fig. 3. Diagram of SVM model. (For interpretation of the references to color in the text, the reader is referred to the web version of this article.)

(shown in red) is calculated by the largest possible distance between the closest points of each data section [54]. This is known as the “margin” and represents the distance between the dashed lines in Fig. 3.

The hyperplane can be represented algebraically by Eq. (16) [53].

$$f(\boldsymbol{x}) = \mathbf{w}^T\boldsymbol{x} + b \tag{16}$$

Eq. (17) represents a training hyperplane where the x values are the examples closest to the hyperplane. The values closest to the hyperplane are known as support vectors, and the representation of this training hyperplane is known as the canonical hyperplane [53].

$$|\mathbf{w}^T\boldsymbol{x} + b| = 1 \tag{17}$$

The distance between a point x_i and the hyperplane, r_i, can be calculated by Eq. (18).

$$r_i = \mathbf{y_i}\frac{\mathbf{w}^T\mathbf{x_i} + b}{|\boldsymbol{w}|} \tag{18}$$

From Fig. 3 above, the margin can be shown as the twice the distance to the support vectors. The margin is shown by Eq. (19).

$$\boldsymbol{M} = \frac{2}{|\boldsymbol{w}|} \tag{19}$$

And finally to maximize margin $\boldsymbol{w}$ and b need to be determined such that $\boldsymbol{M}$ is maximised and for all $(\boldsymbol{x_i}, \boldsymbol{y_i}) \in \mathbb{D}$, $\mathbf{y_i}(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$.

It should be noted that SVM requires more training to produce accurate results, and it is only as accurate as NB for classification over large datasets [37]. For a simple operation such as sentiment determination with only positive and negative classes, NB is preferred [44]. For a large dataset requiring multiple classifications, SVM may provide greater accuracy [52].

Machine Model of sentiment analysis

The section below details a computational system (referred to as “Machine Model”) used to analyze OPRs. The aim of the model is to pair sentiment words including adjectives, verbs, and adverbs with a product feature; typically, a noun. For example, “bad” with “quality”, “durable” with “fabric” or “useful” with “cooler.” These are known as Feature-Sentiment Pairs (FSPs). Fig. 4 illustrates a six-step process by which computers can automatically generate FSPs.

Step (1): tag word types via Part-of-Speech (POS) Tagger

The POS Tagger determines which words are nouns, adjectives, adverbs, and verbs. Other word types such as pronouns are not considered due to their non-interpretability when taken out of context. Also, WordNet only classifies synsets for nouns, adjectives, adverbs, and verbs. Product features in a sentence are mostly nouns, so assume that all nouns in each sentence refer to product features. There are many exceptions where nouns refer to places or objects irrelevant to the product, though these are usually unique and can be excluded from data once the frequency of nouns is collated.

Similarly, assume all verbs, adverbs, and adjectives refer to the sentiment of a sentence. Verbs describe the context or use cases, and can alert designers to whether a part is operating correctly or failing, or whether a customer “likes” or “hates” a feature.

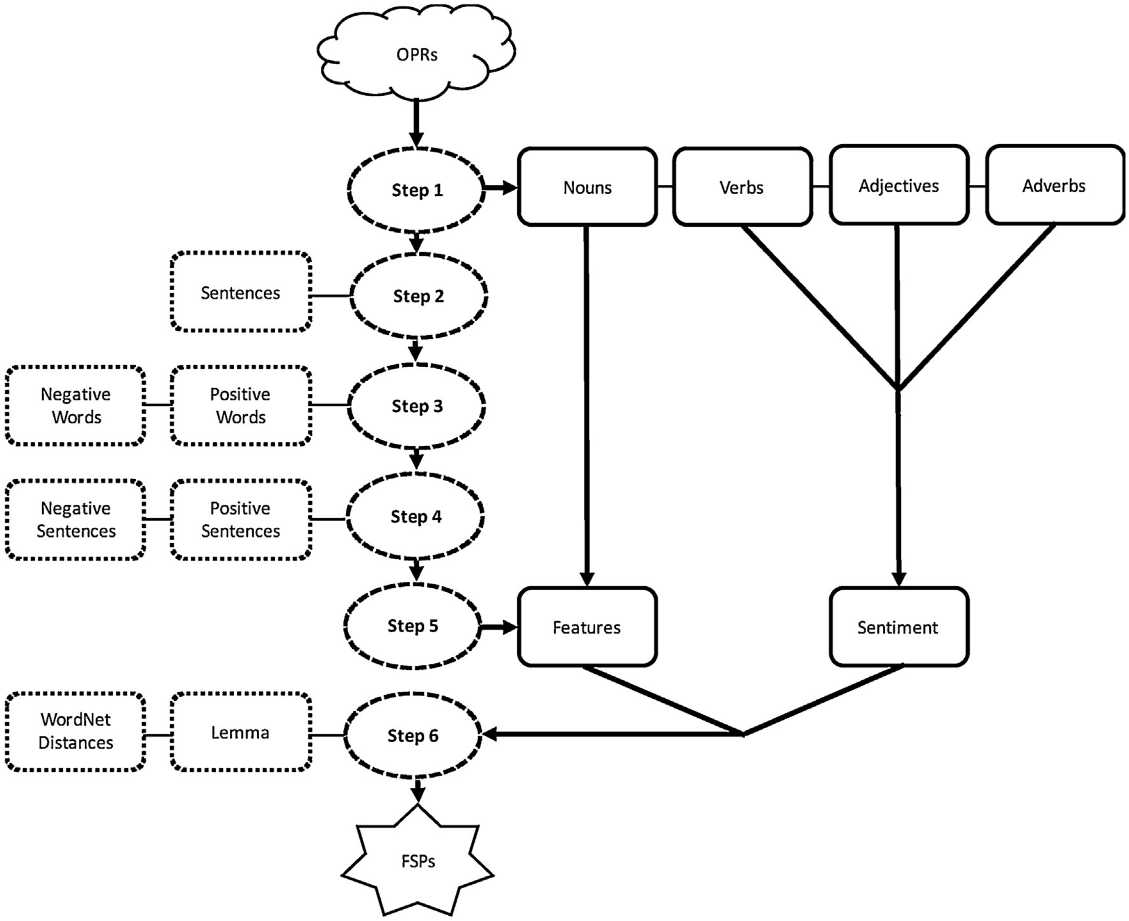


Fig. 4. Diagram of Machine Model Process.

Adjectives and adverbs are by their nature descriptive and therefore are key to determining a sentence's sentiment. For example, adjectives like "comfortable" or "broken" clearly convey a positive or negative sentiment. Adverbs such as "beautifully" and "catastrophically" do the same.

The Stanford POS Tagger can be incorporated into Java and Python languages. Online POS Taggers can also be used to process individual reviews [55,56], from which the reviews can then be analyzed. The tagger will automatically group words into a 2–3 letter acronym (POS Tag) denoting the word type from the Penn Treebank Project [57]. All nouns, verbs, adjective, and adverbs are then categorized into separate arrays according to their tag. The tagger considers the tense and plurality of each word, so the following assumptions in Table 6 can allow for word types to be grouped easily. From here on Nouns equate to "Features" and Verbs, Adverbs and Adjectives collectively as "Sentiment."

Step (2): divide a whole review into separate sentences

In an ideal computational model, the code is directly sourced from Amazon.com itself. The total number of reviews for analysis is denoted as "n". The text can then be processed in 3 steps as shown in Fig. 5: (1) Convert all text in review array R_n into lower case to minimise variance caused by case sensitivity, remove and replace punctuation ".", "!", "?", ";" and ":" with "," to simplify the separation of reviews into sentences and remove punctuation such as "(", ")", "/", "-" as these do not delimit sentences; (2) Create new sentence arrays S_n from each cell in R_n using "," as a delimiter; (3) Identify the nouns (n_n), verbs (v_n), adjectives (aj_n) and adverb (ad_n). The tree-like structure breakdown of word types shown in Fig. 5 is adapted from the Opinion Observer NLP model prescribed by Liu et al. [22].

An example of a Review cell (R_1) separated into a Sentence array (S_n) is shown in Table 7. Note that the example in Table 7 is a genuine OPR sourced from Amazon, and therefore subject to some limited clarity and diction.

Step (3): train the model to predict sentiment

There are many different machine processes for measuring the sentiment of sentences and words. The Naïve Bayes (NB) algorithm is selected for this purpose as it is very accurate even with small datasets. NB needs to be trained from previous data to make accurate predictions based on prior probabilities. Training the algorithm can involve using an existing database of reviews that are already labeled manually, or using words that are already assigned a sentiment. This method requires a large amount of manually entered data to train the NB algorithm.

Table 6
Assumptions for POS tags.

Word types	POS tag	Assumption
Nouns	NN (noun singular) NNS (noun plural) NNP (noun proper) NNPS (noun proper plural)	All tags with "NN" = nouns
Verbs	VB (verb base) VBD (verb past tense) VBG (verb present participle) VBN (verb past participle)	All tags with "VB" = verbs
Adjectives	JJ (adjective) JJR (adjective comparative) JJS (adjective superlative)	All tags with "JJ" = adjectives
Adverbs	RB (adverb) RBR (adverb comparative) RBS (adverb superlative)	All tags with "RB" = adverbs

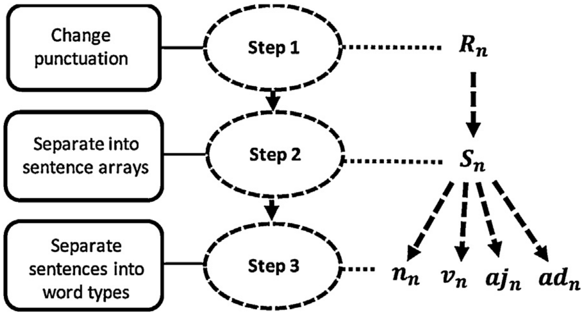


Fig. 5. Tree diagram of array structure.

A simplified approach to train the NB algorithm is to assume that on average sentiment words in 1–2 star reviews are of negative sentiment and those in 4–5 star reviews are of positive sentiment. Assume 3-star reviews to be neutral, and not to be used to train the algorithm. This assumption is based on the strong correlation between product ratings and qualitative sentiment of the product review by Decker and Trusov [58]. Furthermore, Cheung et al. established a direct link between product rating and sentiment towards product attributes [59]. There may be occasions where negative words are used in 5-star reviews, however, such discrepancies are accounted for by analyzing sentiment of sentences in Step 5 (Section "Step (5): generate Feature-Sentiment Pairs from sentences").

As NB operates by probability, the total number of positive and negative words need to be calculated. The more frequent occurrence determines the sentiment, and identical frequencies are deemed neutral.

Step (4): determine sentiment of sentences

Once trained, NB is then used to evaluate the sentiment of each word and sentence based on prior probabilities. As it encounters new words it becomes increasingly more accurate. NB was selected as algorithms such as Maximum Entropy and SVM require significant training and large datasets to become accurate. Apriori is an efficient algorithm for deriving associations from existing data, however, not as effective in determining sentiment [26]. NB is ideal for binary classifications such as sentiment and hence was chosen for this framework.

In an ideal case, NB would be able to self-learn as it analyzed the data and analyze all words independently to determine sentiment; for example, the sentiment of each word is configured individually, not collectively. Thus, a limitation of independence is an inability to recognize common patterns, such as discovering if certain negative words are associated with features. Drawing from Section "Naïve Bayes classifier", Eq. (20) shows the process of determining the sentiment of each sentence.

$$\begin{aligned}\textit{Sentiment of sentence}(+) = {} & (\textit{Probability sentiment is}+) \\ & \times \prod \{\textit{Probability of a word is}+\}\end{aligned} \tag{20}$$

For a small dataset, the sentiment of each sentence can be based on the total probability of each case rather than through continual learning. For extremely large and changing datasets, an evolving NB algorithm is pertinent to maintain accuracy.

Another assumption in this model is that only adjectives, adverbs and verbs from each sentence are to be used to calculate a sentence's overall sentiment. Usually, due to NB's independence, it would consider all words including pronouns, conjunctions, and words that generally do not individually contain sentiment. With a small dataset, words intended to have no sentiment, such as "I", "but", "it" and "they", may affect the overall sentiment based on the assumption that the sentiment of words was calculated only from the rating. It is possible that sentences that contain these words

Table 7
Review to sentence breakdown example.

R_n	Review array	S_n	Sentence array
R_1	*i was looking for a chair that would be easy to carry strong enough to hold me and with at least one little accessory, my husband and i are both big people and i needed a chair that would hold us sitting upright and not low to the ground so we can sit comfortably and watch our grandkids ball games, we are pushing 60 and overweight this chair is very comfortable, it has not only a cup holder but a small mesh flap that will hold a cell phone attached to that another flap for a book and/or magazine and on the opposite side a small cushioned "cooler" pouch that will hold at least 2 soda cans or water bottles, these chairs are great so glad i ordered 2,*	S_1	i was looking for a chair that would be easy to carry strong enough to hold me and with at least one little accessory
		S_2	my husband and i are both big people and i needed a chair that would hold us sitting upright and not low to the ground so we can sit comfortably and watch our grandkids ball games
		S_3	we are pushing 60 and overweight this chair is very comfortable
		S_4	it has not only a cup holder but a small mesh flap that will hold a cell phone attached to that another flap for a book and/or magazine and on the opposite side a small cushioned "cooler" pouch that will hold at least 2 soda cans or water bottles
		S_5	these chairs are great so glad i ordered 2

may have a stronger sentiment than others. For example, negative reviews have a higher frequency of "I" than positive reviews. As the focus is on features and separate sentiment words this statistical phenomenon was not considered vital.

This model also disregards the negation of words; thus, the word "not" did not affect sentiment. It is possible to create functions to detect negation, however, the use of such negations is relatively insignificant compared with common sentiment words.

Based on these assumptions, Eq. (20) above becomes more like Eq. (21).

$$Sentiment\ of\ sentence(+) = \prod \left\{ \begin{matrix} Number\ of \\ +key\ words \end{matrix} \right\} \tag{21}$$

Step (5): generate Feature-Sentiment Pairs from sentences

Once each sentence is classified as positive or negative, Feature-Sentiment Pairs (FSPs) are formed by pairing the sentiment words (adjectives, adverbs, and verbs) with their respective features (nouns). Assume that all adjectives, adverbs, and verbs apply to all the nouns used in a sentence. For example, if two nouns are mentioned in the same sentence, all the sentiment words apply to both nouns. Despite the possibility that sentiment words may apply to incorrect nouns, the small size of the sentence arrays ensures that over an average the most important FSPs will be identified. Fig. 6 represents how FSPs are formed based on this assumption. The resultant output of this step involves clusters of either positive or negative sentiment words around a noun.

Step (6): determine the Importance of Feature-Sentiment Pairs

The output of Step (5) can involve the creation of thousands of FSPs. Thus, the aim of Step (6) is to identify similarities between these FSPs and then measure their frequency to determine the most statistically significant FSPs. This process is shown in Fig. 7.

The process involves four key steps: (1) Simplify words into their lemma using the PlingStemmer program; (2) Use WordNet to identify words within two WordNet "nodes" of each other and assume these to be similar as per Section 3.2.1; (3) Pair similar nouns with similar sentiment words to collate like FSPs, assuming the most frequent terms encompass all similar words; (4) Count the frequency of each distinct FSP. The resultant output is a frequency of FSPs, which gives designers quantitative information to make more informed decisions.

Case study

Raw data for analysis

A case study was conducted to validate the Machine Model of sentiment analysis proposed above. The product selected for analysis was the "Coleman Oversized Quad Chair with Cooler" (Coleman Chair hereafter) [60]. Fig. 8 below shows the provided photo of the product on the Amazon link, with its main features labeled and the numerical data on its OPRs. The Coleman Chair is a foldable and transportable chair that is typically used in an outdoor environment.

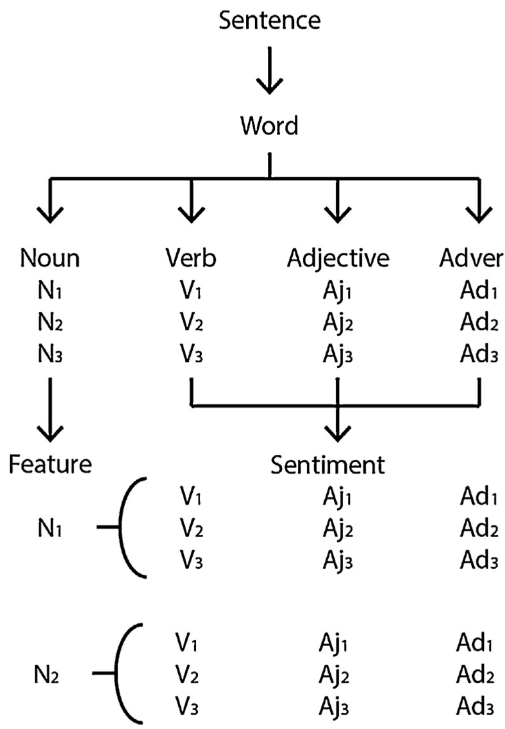


Fig. 6. Diagram of FSPs.

The top 10 "most helpful" reviews for each rating (i.e., 5 star, 4 star, 3 star, 2 star, and 1 star) were analyzed; consisting of 50 reviews in total. Despite 71% of the reviews being 5 stars, an equal quantity of reviews from each rating was used to analyze a broad range of sentiments as it is important for a designer to understand both positive and negative sentiments towards features.

The Machine Model methodology is designed to be versatile and applicable to a range of products. However, this specific product was selected for the case study for several reasons. First, the chair consists of identifiable and separate product features, which enables a machine to identify these features and alert designers to sentiment felt towards them. Secondly, the product enables designers to incorporate elements of both user experience and technology driven design. For example, whether customer finds it comfortable or which parts of the product are failing. Finally, the product contains a large volume of reviews with a variety of positive and negative reviews to consider.

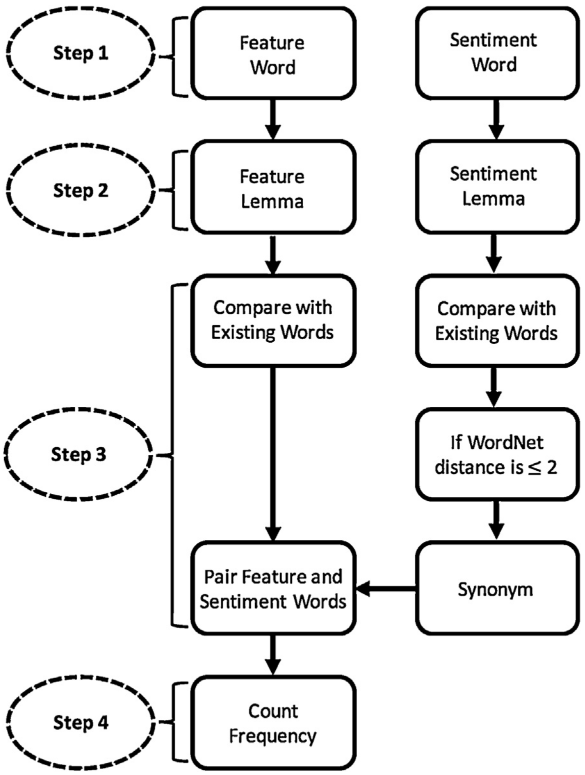

Step 1
Feature Word
Sentiment Word
Step 2
Feature Lemma
Sentiment Lemma
Compare with Existing Words
Compare with Existing Words
Step 3
If WordNet distance is ≤ 2
Pair Feature and Sentiment Words
Synonym
Step 4
Count Frequency

Fig. 7. Process for collating like FSPs.

Human Model of sentiment analysis

Since the objective is to train the machine to think like a human designer, we also developed a Human Model as a control (as illustrated in Fig. 9). The model is abstracted based on observations of how a human designer performs sentiment analysis. The designer was tasked to analyze the same set of 50 reviews, and the results were compared against that of the Machine Model. A conscious decision was made to allow the designer to use their own cognition to develop FSPs within the boundaries of pairing a sentiment with a feature. The resultant output was more varied than the Machine Model. Analysis of these differences, therefore, offers greater insight into the disparity between designer cognition and machine computation.

Step (1): interpret data

The designer first read each of the reviews. Here an underlining assumption is that the designer fully understood the meaning, type, and context of each word for every sentence. The designer also identified synonymous words, their sentiment, negation, irony, and degree of emphasis. These assumptions are considered valid due to the designer's adept knowledge of the English language and the simplicity of language used in the OPRs.

Step (2): infer patterns from data

Once each sentence is understood, the designer inferred a series of patterns [59,61]. For example, consider two comments from separate reviews: “the material is durable” and “excellent, hard-wearing fabric.” From the designer's own knowledge, they understood that “material” connoted the word “fabric”, and thus induced these to be identical. Similarly, the terms “durable” and “hard-wearing” are undisputed synonyms; therefore, by deductive inference they are the same. By repeating the same process of inference as above, associations between features and sentiment can be extracted from the reviews.

From the designer's own knowledge and cognitive processes, similar sentiments were associated with similar subjects. Thus, the designer can formulate FSPs. Unlike the Machine Model, the designer used a high level of inference to identify similarities between alternatively worded phrases. For example, the designer can determine that “the cooler is fantastic,” “the cooler is able to hold loads of beer!” and “I like the cooler” are all positive sentiments of the cooler design. The designer chose to group these comments into the category of “cooler/useful” to simplify the results.

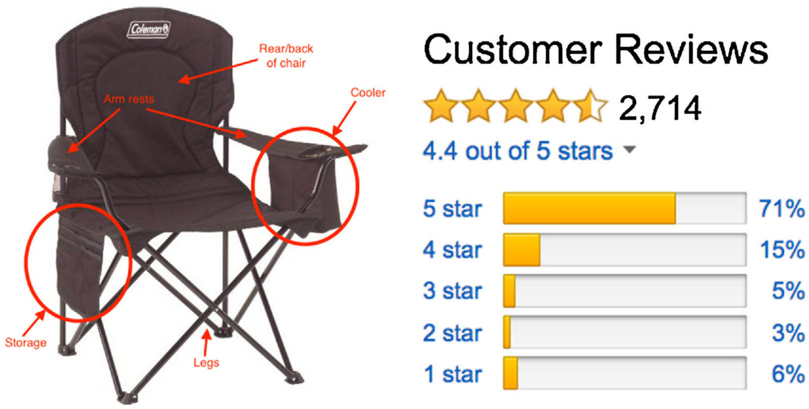
The degree or emphasis of each review's sentiment was not measured, as this introduces too many variables and greater subjectivity to a simplified design process. For example, no distinction was made between “I absolutely LOVE the cooler” and “the cooler is pretty cool,” despite a greater sentimental emphasis. Both were categorized as “cooler/useful.”

Step (3): measure frequency of Feature-Sentiment Pairs

The most simple and objective method for determining the significance of each FSP is their frequency. By repeating Steps 1 and 2, patterns of FSPs emerged. Each time an FSP was inferred, if it was associated with another its frequency was counted.

Step (4): select Feature-Sentiment Pairs based on highest frequency

The most frequent FSPs are then used to determine the CNs for the product. Frequency of FSPs allowed for a quantitative and objective measure of sentiment significance. The alternative would involve weighting comments based on emphasis, however, this would introduce an undesirable level of subjectivity to the results. The most important FSPs can then be used to manually generate CNs.



Customer Reviews

★★★★☆ 2,714

4.4 out of 5 stars

5 star	71%
4 star	15%
3 star	5%
2 star	3%
1 star	6%

Fig. 8. Photo and customer reviews of Coleman chair.

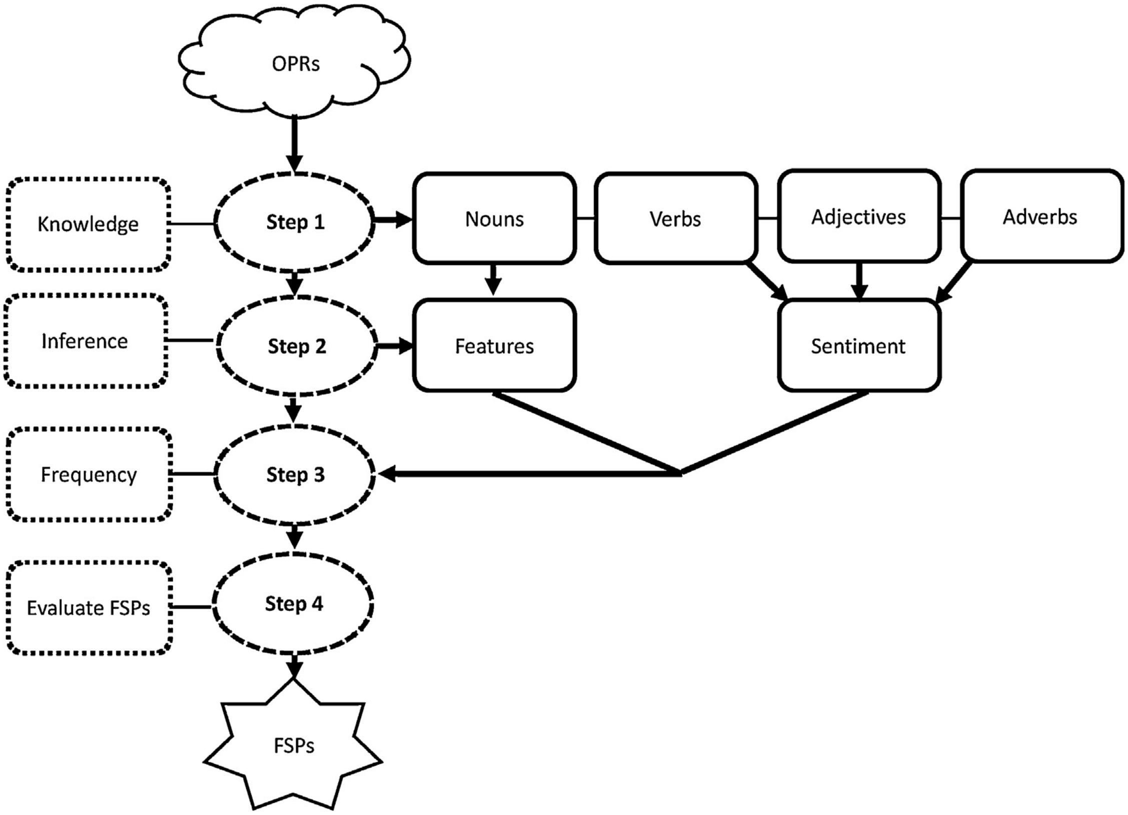


Fig. 9. Cognition process of the designer.

Results of sentiment analysis

Results of the Machine Model

The results of the Machine Model case study are shown in Figs. 10 and 11. From Fig. 10, the most frequent positive FSPs were that the chair is comfortable, generally liked, that it is sturdy and that the cooler and pouch were excellent features. From Fig. 11, the most frequent negative FSPs were that the arms broke, the chair is comfortable, the chair needs to be better, the chair is too large, and that the chair fails.

As shown in Fig. 10, the large size, comfort, sturdiness, cooler and pouch were all regarded as positive features. The results show that the chair itself is positively regarded and appreciated by customers, and the information generated assures designers of the importance of some features and quality design. Fig. 11 shows that the main problems were concerning the arm rests, support and the legs broke easily. This suggests that there are some serious structural issues associated with the chair that if improved would satisfy more customers.

Results of Human Model

Following the methodology explained above the designer manually correlated FSPs by reading the online reviews. Figs. 12 and 13 show the frequency of positive and negative FSPs, respectively.

As illustrated in Fig. 12, the positive FSPs were that the cooler is a useful feature, the chair is comfortable to sit in, sturdy and suitable for tall people. Hence, a designer can identify the structural integrity and comfort as needs to then later “map” functional requirements (FRs) to. Distinguishing positive FSPs is an important part of the process, as some may need to be weighed up against those who regard these positive features as negative. In this study, the significance of sentiment depends on FSP frequency.

As illustrated in Fig. 13, the most common negative FSPs are that the chair is not suitable for a heavy person, that the arm rests are “weak”, that the bag is poor quality and that the chair is poor quality. Interestingly, customers regard the chair paradoxically as being both structurally sound and inept. Therefore, a designer can deduce that a strong CN exists to make the chair more structurally sturdy.

Discussion of results

Achievements of framework design

The utility of the Machine Model demonstrates the strong potential for data-driven design. The similarity of the Machine and Human Models, the accuracy of the model and its applicability to design decision-making are significant achievements of the framework.

One of the most notable achievements of the framework design is the parity between the Machine Model and the Human Model. The Machine Model’s identification of positive chair features such as sturdiness, size, comfort, and practicality of the cooler correlated with the Human Model. The propensity for the chair – especially its arms – to break under load was also identified as the primary negative features in both models. The accuracy of the Machine Model demonstrates that it can sufficiently transform unstructured data into useful information.

The Machine Model was able to generate reasonably meaningful results despite the relatively small dataset of 50 reviews. Data analysis, particularly when using machine learning

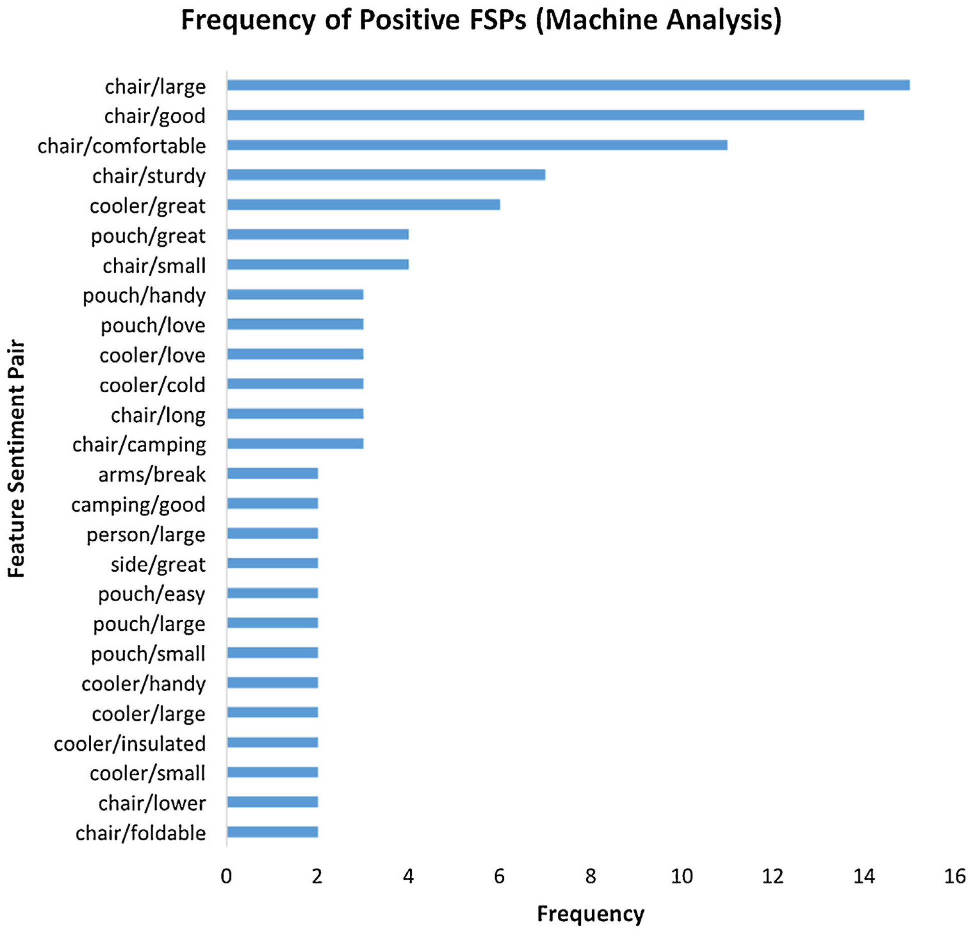


Fig. 10. Frequency of positive FSPs for Machine Model.

techniques, becomes more accurate with large datasets (i.e., a dataset of more than one thousand reviews). Therefore, given the efficacy of this current model on a small dataset, it is highly likely that it would produce more accurate results if applied to a larger dataset.

The present model serves as the first step in validating this method of OPR data-driven design. Its true significance will arise when the framework is applied to bigger datasets. For example, if the Model analyzed thousands of reviews, including those of similar products, it could allow for designers to pinpoint what specific features are lacking in their present products, what differentiates superior substitutes, and quantify the importance of different features for the purpose of design analysis.

The framework served as a successful method to condense unstructured data into information designers can use. It achieved an optimal balance between qualitative customer preference and quantitative validation, allowing designers to make fast, data-driven decisions based on large amounts of input data. Presently, without advanced artificial intelligence, it is impossible for machines to replace creative humans in directly translating CNs into FRs. However, the design process can be enhanced by using machines to condense datasets too large for humans to analyze manually.

The framework can be used to collate and monitor OPRs from a range of sources and product types. It can be used to identify trends and measure how product features perform as they are altered in real-time. The framework represents a shift into a data-driven era of design.

Limitations of the framework design

The framework serves as a successful proof-of-concept for using OPRs in data-driven design. However, there are some inherent limitations that should be considered.

Quantitative oversight

The creation of broad, generalized statistical trends provides easily interpretable results. However, the more quantified the data, the less specific the qualitative information on CNs becomes — a phenomenon dubbed "quantitative oversight." By reducing complex sentences to interpretable quantitative information, finer details are excluded from analysis in two key instances. First, the results in Section "Results of the Machine Model" reveal that customers like the chair's large size and dislike the fragility of its arms. However, their causation, such as situations or specific feature characteristics which effectuated the sentiment, remain unknown. The current framework can reveal what designers need to improve, but cannot elucidate upon why customers expressed particular sentiments. Second, quantified results do not address any unique and informative insights generated by customers. These consist of helpful and highly creative suggestions which may provide designers with new ideas. Due to the infrequency and complexity of such comments, the framework precludes these unique insights.

Creating more insightful results

One of the key limitations of the framework is the interpretability of the FSPs. As FSPs are only a pair of words,

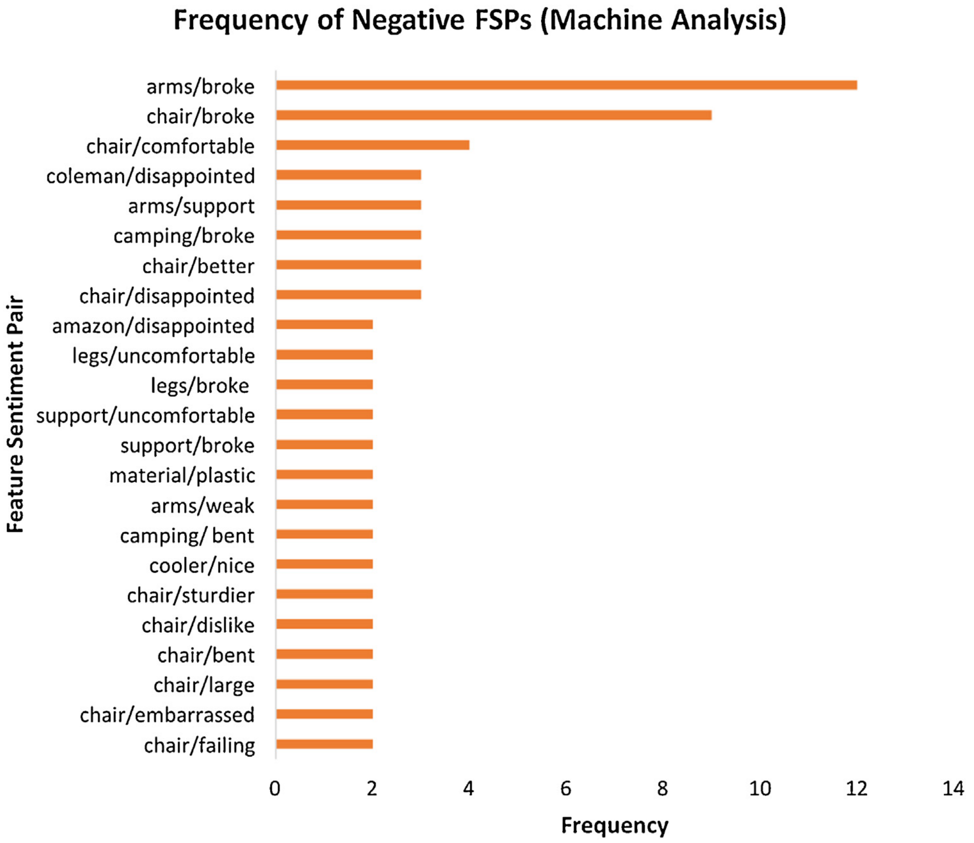


Fig. 11. Frequency of negative FSPs by Machine Model.

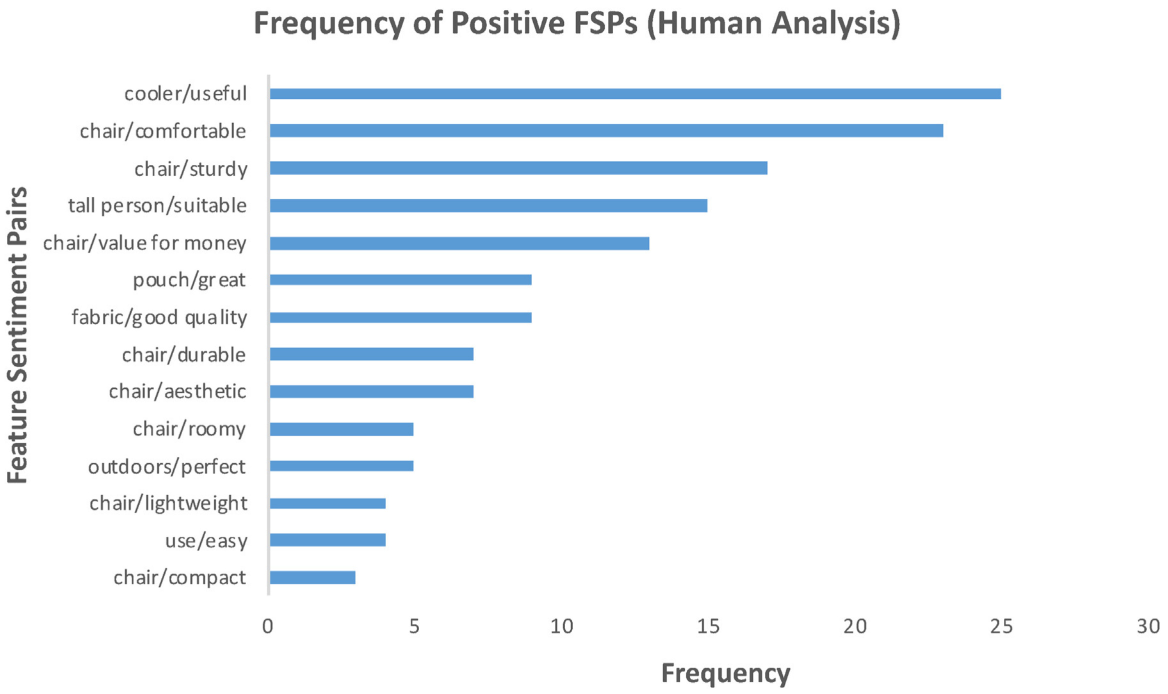


Fig. 12. Frequency of positive FSPs by Human Model.

the lack of detail offers limited insight. For example, one can deduct from the “pouch/great” FSP that the sentiment towards the pouch is positive. However, it cannot reveal why; for example, whether this is due to its size, location on the chair or quality of the material. A possible solution could be extracting a set of the highest quality sentences which contain information on the most important FSPs, similar to Jin et al.’s method of extracting the best sentences [4]. Including more qualitative information could

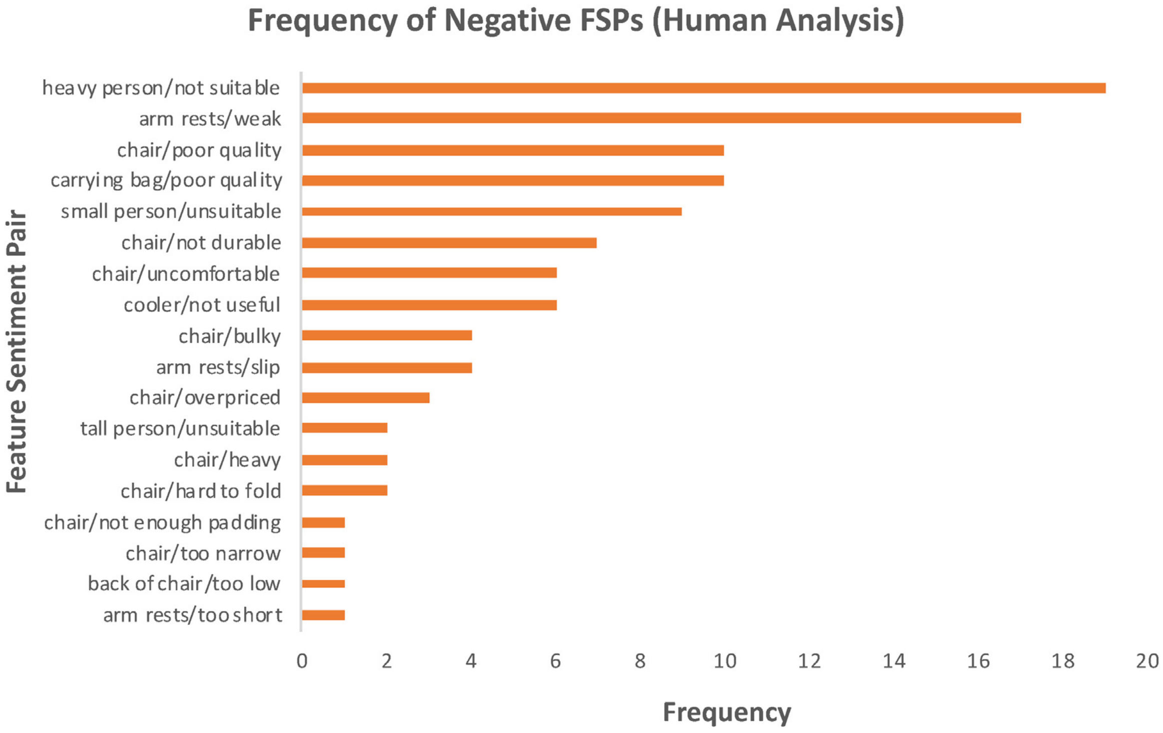


Fig. 13. Frequency of negative FSPs by Human Model.

provide designers with a heightened understanding of why certain features are more popular. Alternatively, more machine learning algorithms such as Apriori, Support Vector Machine (SVM) and Maximum Entropy (ME) can be incorporated to identify even more patterns and trends.

Analyse online product reviews of multiple products

Analysing multiple products of the same type would provide more insightful informasingular product approach tested in this framework. The success of this framework for a single model suggests that a multi-product approach is achievable. An example of future use could involve analyzing many Amazon substitute products to determine what features of competitors' products lead or lag relative to others.

Disparity between Human and Machine Model Feature-Sentiment Pairs

The formulation of FSPs by the Human Model produced slightly alternate FSP combinations than that of the Machine Model. Despite the challenge this represents for evaluating the model's accuracy, it is imperative to highlight the cognitive differences between the human and machine cognition. For example, the designer inferred that people complained of their chair breaking due to the heavy weight of the user. However, the Machine Model only identified the fact the chair broke, but not the cause. Furthermore, this difference can also be accounted for due to the fact each model collated FSPs based on frequency, and that there is a difference between the vocabulary of WordNet and the designer. For future models which aim to apply the model to bigger datasets, it would be important for the human control to be performed within similar constraints to WordNet. For example, the designer can only use single words to describe the sentiment the same as the Machine Model, or collate sentiment based on sentiment words directly paired with features. Depending on the direction of research, inclusion of both a cognitive model and control model could help close the gap between human and machine logic.

Areas for improvement of the Machine Model

Variance of sentences

The greatest single cause of discrepancy and error in the Machine Model is the linguistic variability of OPRs. There are infinite permutations of verbs, adjectives, adverbs, and nouns to describe product features and use cases. Therefore, the Machine Model cannot always infer a relationship between a product's feature and words describing its sentiment. For example, one sentence may talk at length about a specific use case such as "at the ball game I was watching my son play and when I sat down I slightly damaged the legs." Using this model, the term "damaged" may have only been associated with "legs" once, and therefore was not included in the results despite the fact it reveals a clear fault with the strength of the chair's legs. A designer can induce based on their own experience that the legs were weak, though the Machine Model cannot. The limited degree of inference the Machine Model can make relative to the Human Model is shown by the difference in detail between the FSPs in the results.

The assumption of synonyms equating to words within 2 or fewer WordNet nodes (Section "WordNet") ensured a higher degree of accuracy in FSP generation. However, it restricted the Machine Model's ability to infer similarity or relatedness of words. After generating over 200 FSPs, only 60 were used in the final results. One possible solution is to increase the number of WordNet nodes, to allow for a broader interpretation of synonyms. However, it is likely that this might also increase the margin of error and produce less accurate results. Future models must strike a balance between degrees of inference and similarity.

Missing key words in sentences

A comparison between Figs. 10 and 12 and Figs. 11 and 13 (Section "Results of sentiment analysis") reveals how the Machine Model was not as effective as the Human Model in collating the frequency of related FSPs. One cause is that the computational process failed to identify keywords in a sentence, or incorrectly tagged word types. Another cause is that the model detects sentiment words but not the nouns to which it pertains, and vice versa. If either a noun or key sentiment word is missing from a sentence, the sentence cannot generate an FSP. When collating the final results close to 20% of all sentence arrays were disregarded because of this flaw. A possible solution is to employ a strategy similar to Hu and Liu, where the distance between adjectives and nouns is measured [6]. This could improve the Machine Models ability to accurately associate sentiment with features.

Exclusion of pronouns

The use of pronouns such as “it” and “they” to refer to product features is very common in OPRs. Due to the difficulty in making associations with pronouns, they were not counted as nouns, and hence were ignored. Manual analysis of the generated FSPs determined that 10% of all sentiment words were associated with the pronoun “it.” The pronoun was used more often in negative reviews, which may account for the fact that there was fewer machine generated negative FSPs than positive FSPs. A potential solution to this would be making associations of “it” to previous sentences in the same review. This could, however, result in an increased likelihood of error. Furthermore, a stated in the methodology, applied over 100 s or 1000 s of reviews, using only features as nouns is accurate enough.

Exclusion of negation

The negation of an adjective, adverb or verb was not considered in this model, which inhibited its accuracy. For example, the word “not” when used with a descriptive word usually reverses its negation. The use of “not” was uncommon in positive reviews, however, its exclusion affected some of the results obtained from the negative reviews. This may account for the fact there were 3 distinctly positive FSPs in the negative FSP results in Fig. 11 (Section “Results of the Machine Model”) above. Using a method similar to Fang and Zhan’s model, where the use of “not” directly next to a sentiment word reversed its negation, could help solve this issue [31].

POS Tagger accuracy

Another error which significantly impacted the quality of the results was the inaccuracy of the POS Tagger. Though the POS Tagger used is generally accurate, there were a few instances of completely incorrect labeling and errors that resulted from common polysemous words. One example of the polysemous error is the incorrect labeling of the “cooler” feature, which is a term for a device which keeps things cold. However, it is also a common comparative adverb. In this case study, “cooler” nearly always referred to the feature, though the POS Tagger algorithm has been trained on data where “cooler” is usually an adverb. Therefore, in many cases, it does not identify “cooler” as a feature. By the same logic, any use of the word “cooler” as an adverb also becomes difficult for the machine model to discern. Other examples of this include the terms “cold” and “love” being both nouns and adjectives. Up to 60% of comments referring positively to the “cooler” feature were ignored because of this problem; shown by its high importance in the Human Model and low importance in the Machine Model. Such errors could be reduced if a POS Tagger could be tailored to specific datasets; for example, by classifying keywords manually.

Method of FSP generation

The assumption that all sentiment words in a sentence apply to all features was a simple method to associate sentiment words with their features. However, many sentences contain multiple nouns and adjectives, and under this assumption, all the sentiment words apply to all the nouns indiscriminately. For example, one common FSP contained the nouns “chair” and “arms” and the descriptive words “comfortable” and “broke.” The Human Model could infer that “chair” goes with “comfortable” and “arms” with “broken”, though the Machine Model cannot. The incorrect association of sentiment words with features resulted in 15% error margin for incorrect FSPs.

This error is negated by the fact that when analyzing large results, the infrequency of incorrect associations deemed them outliers. Thus, the FSPs with high frequencies were the most accurate. However, with a relatively small dataset, incorrect associations have a greater impact. Furthermore, for a data size containing thousands of reviews, the use of TF-IDF and frequency would limit the impact of such outlier and produce more accurate results. Another method for alleviating the error of incorrect associations is using an “n-gram;” i.e., limiting the sentence arrays to “n” number of words. However, the problem with n-grams is they can exclude keywords if sentences contain more than “n” words, reducing FSP accuracy.

Sentiment accuracy

A source of error also occurred in the determination of FSP sentiment. Due to the small dataset, there were many inconsistencies. Similar to most data science models, a larger quantity of input data enhances the accuracy and insightfulness of the output information. In this model using the Naïve Bayes algorithm on a small dataset on average produced accurate results; with an error margin of 20%. Also, positive reviews generally provided more details than negative reviews, which accounts for the lower frequency of negative FSPs. A cogent example of sentiment error is the negative labeling of the FSP “chair/comfortable” which is clearly positive. Similarly, “arms/break”, a negative FSP, was construed by the Machine Model as positive. A designer can easily infer that these are errors, but it highlights an area for improvement in the Machine Model. A larger dataset and an NB algorithm trained to define positive and negative sentences would help to reduce this error.

Inability to interpret non-literal human expression

Present NLP and sentiment analysis methods are unable to interpret irony, idioms and other non-literal forms of human expression. It may be possible with a large dataset to identify such statements as anomalies; for example, a positive statement contained within a negative review could be determined to be ironic. However, it is difficult to discern between what is an ironic statement or a singular positive statement to the high variability of human expression. Future research on training algorithms to identify such non-literal language would be particularly interesting to this field.

Use of advanced computational methods

The dataset used by the Machine Model was captured manually. For the model to scale, the use of an Application Programming Interface (API) to autonomously capture online data would be required. There is a clear trend that more and more e-commerce platforms are bonding deeper ties with manufacturers in terms of how a product should be designed. For example, NetEase is a pioneering ecommerce platform in China that collaborates closely with OEM to design, manufacture products that meet “stricter selection requirements.” In that regard, API plays a critical role in converging OPRs collected from different courses into the manufacturer’s reach. For those Internet Infrastructure Builders (e.g., Amazon and Alibaba) that own both e-commerce platforms and web services, in order to capitalize the values of data (e.g., online product reviews), it is necessary to provide a more intuitive way to present the data to enterprise customers in a largely customized fashion. For example, it will be mutually beneficial for both parties if manufacturers are guided to use various Amazon Web Services (e.g., Amazon Comprehend) to analyze various data captured on Amazon.com.

Integration of Amazon Comprehend's Sentiment Analysis and Topic Modelling features could provide a platform for analysis of larger data sets without the need for an entire model to be built from scratch [62]. It would provide an efficient process for the Machine Model to pre-capture positive and negative sentences, and from there generate Feature Sentiment Pairs. Furthermore, the Topic Modelling and Entity Recognition features could allow designers to identify broad trends and then focus on data of highest significance. For example, Topic Modelling could be used to determine which features are most commonly found in negative or positive reviews. The Machine Model could then look for all words which are similar to these key features, and process the data to generate Feature-Sentiment Pairs. Amazon Comprehend does not possess the specificity of the Machine Model's insights, though it provides an expedient process for identification of terms that the Machine Model should focus on, and perhaps offer more accurate sentiment analysis.

Areas for improvement of the Human Model

Human inference variability

One difficult-to-measure variable is the level of inference used by the designer to interpret the data. For example, there are many sentences which describe a context to convey sentiment such as "the cooler can fit 10 cans in it!" The nouns and descriptive words alone are not indicative of sentiment, so it requires a human to make the inference based on their knowledge. Thus, there is a strong potential for bias based on the designer's existing knowledge and experiences that may have resulted in the incorrect association; for example, the previous quote could also be interpreted as a criticism. The volume and vagueness of the inferences made it difficult to measure the error of the designer's judgments. To overcome this error, the use of multiple designers to generate an average would reduce the impact of individual assumptions.

One of the key aims of the method was to "map" the inferences of the designer so that a machine can replicate such processes. Due to the infinite number of permutations for human inferences, irony and expressions it is very difficult for a machine to discover trends without an extremely large and sophisticated knowledge base. Currently, machines can identify word types and make associations between individual words, though cannot infer comprehensive meaning from large sentences.

Human error

As the Human Model involves the manual analysis of data by a human, there is a likely occurrence of human-made error. For example, overlooking words, missing sentences, or misinterpretation of definitions. The assumption that the human designer has an omniscient vocabulary and infers the correct meaning is fallible. There may be words the designer does not know, expressions which are unfamiliar or inferences which are based on incorrect knowledge.

It is observed that the human designer might not have processed all the data, overlooked some issues or forgot to tabulate information when collating frequency. They might have also been unconsciously bias, looking for features or sentiments they believe to be most important whilst ignoring those less important. To minimize errors caused by individual bias, it would be ideal to have multiple human designers complete a standardized evaluation of the OPRs and compute an average from the results. Based on the author's previous work of manually analyzing OPRs through the qualitative data analysis process, unless a large group of designers (more than 10 designers) was engaged in the evaluation, it would be difficult to completely eliminate the individual bias. If particular design ontology (Function–Behavior–Structure) must be followed to analyze OPRs, it is very true that the analysis results may vary significantly in accordance to different designers' knowledge and experience. On the other hand, sentiment analysis is a slightly different matter. Sentiment analysis is arguably one of the most frequently performed activities that most people especially designers have been repeatedly trained for years if not decades. That being said, even though it is helpful to engage more designers in the evaluation, it can be argued that one designer is sufficient to produce reasonably convincing results that can be compared with the Machine Model.

Conclusion

The proposed framework applied existing machine learning and computational technologies to OPRs on Amazon.com and produced useful information for designers. The quantitative nature of the information allows designers to evaluate the importance of product features and generate more accurate CNs. This achievement highlights the potential paradigm shift towards data-driven decision making in design. This framework aims to demonstrate both the utility and challenges of data-driven design from online products reviews, such that it can be replicated on a larger scale and transit towards a revolutionary age of design.

With respect to future work, the development of a more powerful and sophisticated computational programs for the Machine Model is pertinent for the framework's application to big data. The framework can then easily analyze thousands of reviews and offer more powerful data-driven insights. Reconstructing a complete model in computational languages as Java or Python could allow for the incorporation of more machine learning algorithms. This is an essential step to apply the framework to multiple products. This study used sentiment analysis techniques derived from NLP methods and Naïve Bayes machine-learning. The application of more complex associative and categorical algorithms such as Apriori, Maximum Entropy and Support Vector Machine may allow for greater statistical insight and accuracy.

Advances in data analytics pave the way for a new paradigm of data-driven product design. The proposed framework exemplifies a specific application — the sentiment analysis of online product reviews. The further expansion of principles developed in this framework could enable designers to quantitatively evaluate the sentiment of a product's features, monitor competing products, evaluate the success of new product features on existing markets and even predict where new design opportunities lie. The overlap between the proposed framework with new evolving technologies is also significant. As the Internet of Things phenomenon offers new ways of tracking the use of products, an even larger pool of data will emerge beyond OPRs, allowing for an even broader application of this framework. It can also apply to new internet trends, or new social media platforms and e-commerce websites as they emerge, producing even more insights.

References

[1] Lee, J., Kao, H., Yang, S., 2016, Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment. Procedia CIRP, 16:3–8. 2–14.

[2] Jin, J., Liu, Y., Ji, P., Liu, H., 2016, Understanding Big Consumer Opinion Data for Market-Driven Product Design. International Journal of Production Research, 54/10: 3019–3041.

[3] Jin, J., Ji, P., Liu, Y., 2014, Prioritising Engineering Characteristics Based on Customer Online Reviews for Quality Function Deployment. Journal of Engineering Design, 25/7–9: 303–324.

[4] Jin, J., Ji, P., Gu, R., 2016, Identifying Comparative Customer Requirements from Product Online Reviews for Competitor Analysis. Engineering Applications of Artificial Intelligence, 49:61–73.

[5] Jian, J., Ji, P., Kwong, C., 2016, What Makes Consumers Unsatisfied with Your Products: Review Analysis at a Fine-Grained Leve. Engineering Applications of Artificial Intelligence, 47:38–48.

[6] Hu, M., Liu, B., 2004, Mining Opinion Features in Customer Reviews. American Association of Artificial Intelligence, 4/4: 755–760.

[7] Hedegaard, S., Simonsen, J., 2013, Extracting Usability and User Experience Information from Online User Reviews. SIGCHI Conference on Human Factors in Computing Systems (Paris).

[8] Suh, N., 2001, Axiomatic Design: Advances and Applications. Oxford University Press.

[9] Tontini, G., 2007, Integrating the Kano Model and QFD for Designing New Products. Total Quality Management, 18/6: 599–612.

[10] Chan, L.K., Wu, M.L., 2002, Quality Function Deployment: A Literature Review. European Journal of Operational Research, 143/3: 463–497.

[11] Matzler, K., Hinterhuber, H.H., 1998, How to Make Product Development Projects More Successful by Integrating Kano's Model of Customer Satisfaction into Quality Function Deployment. Technovation, 18/1: 25–38.

[12] Ertay, T., Büyüközkan, G., Kahraman, C., Ruan, D., 2005, Quality Function Deployment Implementation Based on Analytic Network Process with Linguistic Data: An Application in Automotive Industry. Journal of Intelligent & Fuzzy Systems, 16/3: 221–232.

[13] Purohit, S., Sharma, A., 2015, Database Design for Data Mining Driven Forecasting Software Tool for Quality Function Deployment. International Journal of Information Engineering and Electronic Business, 7/4: 39–50.

[14] Shen, X., Tan, K., Xie, M., 2001, The Implementation of Quality Function Deployment Based on Linguistic Data. Journal of Intelligent Manufacturing, 12/1: 65–75.

[15] Hou, Z., Chen, S., 2005, Regulatory Method for Importance of Customers' Requirements Based on Kano Model. Computer Integrated Manufacturing System, 11/12: 1785–1789.

[16] Ding, X., Liu, B., 2007, The Utility of Linguistic Rules in Opinion Mining. Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

[17] McAfee, A., Brynjolfsson, E., Big Data: The Management Revolution Harvard Business Review, October 2012. [Online]. Available: https://hbr.org/2012/10/big-data-the-management-revolution. [Accessed 17 October 2016].

[18] Netzer, O., Feldman, R., Goldenberg, J., Fresko, M., 2012, Mine Your Own Business: Market-structure Surveillance Through Text Mining. Marketing Science, 31/3: 521–543.

[19] Archak, N., Ghose, A., Ipeirotis, P.G., 2011, Deriving the Pricing Power of Product Features by Mining Consumer Reviews. Management Science, 57/8: 1485–1509.

[20] Manning, C., 2016, Computational Linguistics and Deep Learning. Computational Linguistics, 41/4: 701–707.

[21] Fang, X., Zhan, J., 2015, Sentiment Analysis Using Product Review Data. Journal of Big Data, 2/5: 1–14.

[22] Liu, B., Hu, M., Cheng, J., 2005, Opinion Observer: Analyzing and Comparing Opinions on the Web. Proceedings of the 14th International Conference on World Wide Web.

[23] Lee, T., 2009, Adaptive Text Extraction for New Product Development. ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.

[24] Justeson, J., Katz, S., 1995, Technical Terminology: Some Linguistic Properties and an Algorithm for Identification in Text. Natural Language Engineering, 1/1: 9–27.

[25] Kwong, C.K., Jiang, H., Luo, X.G., 2016, AI-Based Methodology of Integrating Affective Design, Engineering, and Marketing for Defining Design Specifications of New Products. Engineering Applications of Artificial Intelligence, 47:49–60.

[26] Chung, W., Tseng, T., 2012, Discovering Business Intelligence from Online Product Reviews: A Rule-Induction Framework. Expert Systems with Applications, 39/15: 11870–11879.

[27] Jin, J., Ji, P., Liu, Y., Lim, S., 2015, Translating Online Customer Opinions into Engineering Characteristics in QFD: A Probabilistic Language Analysis Approach. Engineering Applications of Artificial Intelligence, 41:115–127.

[28] Qi, J., Zhang, Z., Jeon, S., Zhou, Y., 2016, Mining Customer Requirements from Online Reviews: A Product Improvement Perspective. Information & Management, 53/8: 951–963.

[29] Amazon, Amazon's Top Customer Reviewers, 22.10.2016. [Online]. Available: https://www.amazon.com/review/top-reviewers. [Accessed 22 October 2016].

[30] Liu, Y., Jin, J., Ji, P., Harding, J., Fung, R., 2013, Identifying Helpful Online Reviews: A Product Designer's Perspective. Computer-Aided Design, 45/2: 180–194.

[31] Princeton University, About WordNet., WordNet, 2010. [Online]. Available: http://wordnet.princeton.edu. [Accessed 1 October 2016].

[32] Scriver, A., 2006, Semantic Distance in Wordnet: A Simplified and Improved Measure of Semantic Relatedness. Ontario.

[33] Boyd-Graber, J., Fellbaum, C., Osherson, D., Schapire, R., 2006, Adding Dense, Weighted Connections to WordNet. The Third International WordNet Conference.

[34] Stanford, Stanford Log-linear Part-Of-Speech Tagger, 2016. [Online]. Available: http://nlp.stanford.edu/software/tagger.shtml. [Accessed 20 October 2016].

[35] Pazzani, M., Billsus, D., 2007, Content-Based Recommendation Systems. The Adaptive Web, Springer-Verlag Berlin Heidelberg, Berlin: 325–341.

[36] Max Planck Institut Informatik, Class PlingStemmer, Yago Nago, 2017. [Online]. Available: http://resources.mpi-inf.mpg.de/yago-naga/javatools/doc/javatools/parsers/PlingStemmer.html. [Accessed 8 March 2017].

[37] Adomavicius, G., Tuzhilin, A., 2005, Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, 17/6: 734–749.

[38] Perone, C., Machine Learning: Cosine Similarity for Vector Space Models (Part III), Incognita, 9.12.2013. [Online]. Available: http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/. [Accessed 15 April 2017].

[39] Huang, A., 2008, Similarity Measures for Text Document Clustering. The Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008) (Christchurch).

[40] Wu, H., Luk, R., Wong, K., Kwok, K., 2008, Interpreting TF-IDF Term Weights as Making Relevance Decisions. ACM Transactions on Information Systems (TOIS), 26/3: 1–37.

[41] TFIDF.com, What Does TF-IDF Mean?, [Online]. Available: http://tfidf.com/. [Accessed 1 March 2016].

[42] Merrett, R., 5 Tools and Techniques for Text Analytics, CIO, 18.5.2015. [Online]. Available: https://www.cio.com.au/article/575209/5-tools-techniques-text-analytics/?pp=2. [Accessed 13 March 2017].

[43] Blei, D., Ng, A., Jordan, M., 2003, Latent Dirichlet Allocation. Journal of Machine Learning Research, 3:93–1022.

[44] Wang, S., Manning, C., 2012, Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. 50th Annual Meeting of the Association for Computational Linguistics: Short Papers.

[45] Berger, A.L., Pietra, V.J.D., Pietra, S.A.D., 1996, A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics, 22/1: 39–71.

[47] Osborne, M., 2002, Using Maximum Entropy for Sentence Extraction. Proceedings of the ACL-02 Workshop on Automatic Summarization.

[48] Nigam, K., Lafferty, J., McCallum, A., 1999, Using Maximum Entropy for Text Classification. In IJCAI-99 Workshop on Machine Learning for Information Filtering.

[49] College of Science and Engineering, University of Minnesota, Association Analysis: Basic Concepts and Algorithms, [Online]. Available: https://www-users.cs.umn.edu/∼kumar/dmbook/ch6.pdf. [Accessed 17 April 2017].

[50] Ng, A., Association Rules and the Apriori Algorithm: A Tutorial, Ministry of Defence of Singapore, [Online]. Available: http://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html. [Accessed 19 April 2017].

[51] Lee, T., 2007, Needs-Based Analysis of Online Customer Reviews. The Ninth International Conference on Electronic Commerce.

[52] Joachims, T., 1998, Text Categorization with Support Vector Machines: Learning with Many Relevant Features. European Conference on Machine Learning: ECML-98, 137–142.

[53] Ray, S., Understanding Support Vector Machine Algorithm from Examples (along with code), 6.10.2015. [Online]. Available: https://www.analyticsvidhya.com/blog/2015/10/understaing-support-vector-machine-example-code/. [Accessed 2 April 2017].

[54] Manning, C.D., Raghavan, P., Schütze, H., 2008, Support Vector Machines: The Linearly Separable Case. Cambridge University Press . [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/support-vector-machines-the-linearly-separable-case-1.html. [Accessed 2 April 2017].

[55] SMILE, SMILE Text Analyzer, 2017. [Online] Available: https://smile-pos.appspot.com/. [Accessed 5 April 2017].

[56] Schram, S., POS Tagging, 2017. [Online] Available: https://parts-of-speech.info/ . [Accessed 18 March 2018].

[57] Penn Treebank Project, Alphabetical List of Part-of-Speech Tags used in the Penn Treebank Project, 2003. [Online] Available: https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html. [Accessed 17 March 2018].

[58] Decker, R., Trusov, M., 2010, Estimating Aggregate Consumer Preferences from Online Product Reviews. International Journal of Research in Marketing, 27/4: 293–307.

[59] Cheung, K., Kwok, J., Law, M., Tsui, K., 2003, Mining Customer Product Ratings for Personalized Marketing. Decision Support Systems, 35/2: 231–243.

[60] Amazon, Coleman Oversized Quad Chair with Cooler, 2016. [Online]. Available: https://www.amazon.com/Coleman-Oversized-Quad-Chair-Cooler/product-reviews/B0033990ZQ/ref=cm_cr_getr_d_paging_btm_next_3?ie=UTF8&showViewpoints=1&sortBy=helpful&pageNumber=3. [Accessed 13 September 2016].

[61] Douven, I., Abduction, Stanford Encyclopedia of Philosophy, 21.4.2017. [Online]. Available: http://plato.stanford.edu/entries/abduction/#DedIndAbd. [Accessed 19 June 2017].

[62] Amazon Web Services, Amazon Comprehend: Developer Guide, 2018. [Online]. Available: https://docs.aws.amazon.com/comprehend/latest/dg/comprehend-dg.pdf. [Accessed 5 May 2018].