Identifying helpful online reviews: A product designer's perspective

Ying Liu a, *, Jian Jin b, Ping Ji b, Jenny A. Harding c, Richard Y.K. Fung d

a Mechanical Engineering, National University of Singapore, Singapore 117576, Singapore
b Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China
c Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, United Kingdom
d Department of Systems Engineering & Engineering Management, City University of Hong Kong, Hong Kong

A R T I C L E   I N F O

Article history:
Received 7 September 2010
Accepted 16 July 2012

Keywords:
Product review analysis
Voice of the customer
User requirement
Opinion mining
Product review
Review recommendation
Product design
Conceptual design

A B S T R A C T

Large amounts of online reviews, the valuable voice of the customer, benefit consumers and product designers. Identifying and analyzing helpful reviews efficiently and accurately to satisfy both current and potential customers’ needs have become a critical challenge for market-driven product design. Existing evaluation methods only use the review voting ratios given by customers to measure helpfulness. Due to the issues such as viewpoints of interest, technical proficiency and domain knowledge involved, it may mislead designers in identifying those truly valuable and insightful opinions from designers’ perspective. Thus, in this study, we initiate our work to explore a possible approach that bridges the opinions expressed by consumers and the understanding gathered by designers in terms of how helpful these opinions are. Our ultimate research focus is on how to automatically evaluate the helpfulness of an online review from a designer’s viewpoint entirely based on the review content itself. We start our work by first conducting an exploratory study to understand the fundamental question of what makes an online customer review helpful from product designers’ viewpoint. Through our study, we propose four categories of features that reflect designers’ concerns in judging review helpfulness. Based on our experiments, it reveals that discrepancy does exist between both online customer voting and designers’ rating. Furthermore, for the cases where review ratings are not steadily available, we have proposed to use regression to predict and interpret review helpfulness with the help of the aforementioned four categories of features that are entirely extracted from review content. Finally, using review data crawled from Amazon.com and real ratings given by design personnel, it demonstrates the effectiveness of our proposal and it also suggests that helpful product reviews can be identified from a designer’s angle by automatically analyzing the review content. We argue that the study reported is able to improve designer’s ability in business intelligence processing by offering more targeted customer opinions. It highlights the urgency to uncover sensible user requirements from such quality opinions in our future research.

1. Introduction

Product reviews are an increasingly important type of user-generated online content since they offer valuable information that helps product designers better understand the needs and preferences of consumers, and in the meantime, influences potential consumers in their purchase decision making [1].

Starting from 2003–2004, there had emerged a stream of research efforts on analyzing such online reviewers [2,3]. It is now known as opinion mining and the major research focuses are product feature identification and sentiment analysis. In the research community, it is generally agreed that such product reviews refer to the free texts written by consumers, rather than domain experts, and are published mainly in e-commerce websites like Amazon.com, websites dedicated to hosting consumer comments like eOpinion.com and user blogs. Online reviews are generically different from classic survey data and data gathered from questionnaires or interviews. These reviews are written entirely based on the willingness of consumers, from their angle of interests, in their language and without any prescribed questions to lead them. Furthermore, online product reviews often vary greatly particularly in terms of how customer opinions are being expressed. Some reviews may contain few words but are centered on the user preferences with respect to several crucial product features, while others are lengthy with only a few sentences containing opinions on product usability. Also, the sheer amount of online reviews available, their geographic locations, reviewer age concerned, and user blogging patterns, have surpassed the capability that currently most industrial companies and marketing firms are able to handle. To what extent that designers can understand customer wants and
preferences is a crucial factor in product design, especially in the conceptual design stage. Fig. 1 illustrates the snapshot of a typical product review of a digital camera.

At the center of the problems discussed above is the core issue concerning the quality of online reviews. As a matter of fact, the quality of information available in a community is often observed to be inversely related to the size of its membership [4]. There exist some limited works about evaluating online review helpfulness. In these efforts, the quality of online review is defined in terms of online helpfulness voting ratio, $x/y$ (x out of y people find a particular review helpful, e.g., “85 out of 88 people found the following review helpful” shown in Fig. 1). This helpfulness voting ratio is regarded as the golden criterion in defining the helpfulness of product reviews. Meanwhile, as consumers are not obligated to vote such reviews, usually, only a small proportion of the reviews eventually receive sufficient votes. It has come to our attention that existing efforts only consider the helpfulness value of product reviews from the consumers’ standing. There exists a visible gap in that helpfulness is not perceived, defined and evaluated from the domain users’ point of view, such as product designers and manufacturing engineers. Do consumers view online product reviews in the same dimensions as designers and engineers do? Can we find a way to automatically filter such a large number of online reviews in a way that designers and engineers consider useful to their design and engineering work?

In this study, we start from a comprehensive user study of review helpfulness conducted with the assistance of design personnel. We had deliberately not provided rating guidelines to the designers who actually acted as review annotators, and they were told that they only need to rate the review helpfulness based on their own design experience or needs. The 1000 online reviews given in the study were randomly chosen from Amazon.com. We adopted a five-degree helpfulness evaluation metric which only concerns whether it is helpful or not helpful towards product design. After the manual annotation, two questionnaires were followed up. According to our analysis associated with the questions asked, we have gained several insights regarding why certain reviews are perceived as helpful by designers, while others are not. Such understanding enables us to propose four categories of features in modeling the concept of product review helpfulness, including linguistic features, product features, features based on information quality and features using information theory. Furthermore, since we understand that human annotation may not be always available, we extend our work to explore whether helpfulness can be modeled entirely based on the features that are derived completely from the review content without referring to any external domain knowledge, rules, human interpretations, etc. To proceed, we examine whether domain features, one of the four categories of features proposed which needs certain domain knowledge like product structure, do provide significant contributions in modeling, or if they can be neglected without a significant loss. An extensive study with many prevailing machine learning algorithms has been conducted using regression and classification in order to test and validate our proposal. To the best of our knowledge, we have not encountered any efforts that target similar issues.

The contributions of this work are at least threefold. Firstly, our approach is fundamentally different from those reported in the existing works. A crucial question on how designers perceive, define and evaluate review helpfulness has not been answered, but it is urgently needed. As suggested from the experimental study, we have observed that there is no strong correlation between the helpfulness voting given by consumers and the one rated by product designers. This challenges the foundation of many existing works. Secondly, we initiate the study on how domain users actually perceive and rate the helpfulness issue of online product reviews with respect to their design work. Instead of arbitrarily defining a set of criteria on helpfulness, selecting representative reviews, and giving them to designers for rating, we have taken a reverse approach—a user centric approach, to first conduct a user study and understand how people actually do that. We then start to identify the underlying issues of helpfulness, features needed, ways of modeling it, and how it can be validated. Thirdly, besides the contribution made towards the helpfulness modeling in the perspective of domain users, we have also explored another crucial problem on the significance of the contribution made by domain related features in helpfulness modeling. This effort demonstrates that it is possible to migrate the model discovered to a new product topic without a significant loss in helpfulness ranking. This is particularly desirable when the manually rated reviews are not steadily available for training. It is part of our research ambition to explore whether the concept of helpfulness, being perceived by different domain users, can actually be modeled as domain free, so that the essential model derived can be migrated to other products.

The rest of this paper is organized as follows: Section 2 introduces the related works. In Section 3, we report our exploratory

Fig. 1. One typical customer review.
study on review helpfulness from designers' perspective. Section 4 illustrates the proposed technical approach, describes our efforts on helpfulness modeling, features proposed and several technical considerations including evaluation metrics. Section 5 presents the comprehensive details of our experimental study using real world data and many prevailing machine learning based algorithms, and discusses the results. Section 6 concludes.

2. Related work

2.1. Opinion mining

In the past few years, there has been an obviously rising interest in the topic of automatic parsing and analysis of online reviews in several major research forums. Hu and Liu proposed a method that uses various word features, including occurrence frequency, part-of-speech tags and synonyms set in WordNet. While they called it a summarization, their basic idea was to identify a noun word and its nearest opinion word. Ding et al. proposed a holistic lexicon-based approach. They utilized external evidence and linguistic conventions to identify the semantic orientation of opinions, and later in a further work on product entity discovery and entity assignment. Recently, a lexicalized hidden Markov model-based learning framework has also been reported. Other related studies include sentiment analysis and subjective classification, for example, using a word sentiment classification approach, a latent semantic analysis-based approach where the cosine distance is introduced, and some syntactic information in the feature sets of a support vector machine. All of these contributions investigate ways to classify a review as positive, negative, or neutral at different levels or from particular perspectives, namely, the word level, the sentence level, the document level and the feature level.

Feature extraction has also been studied by researchers in the last few years in data mining and natural language processing (NLP) communities. Many techniques search for statistical patterns, including words and phrases that appear frequently in product reviews. For example, Hu and Liu utilized the association mining algorithm to generate a set of frequent nouns or noun phrases as possible product features and pruned those unlikely features. Alternative similar techniques used some heuristic frequent language rules to find words or phrases that match the rules patterns and considered these as product features. But, all these methods involve human intervention to some extent. In this paper, we extract the product features using a document profile (DP) model reported in our previous work. The DP model extends the concept of classical Apriori algorithm to discover word frequency patterns automatically and involves the least human intervention, together with the help of part-of-speech (POS) tags to annotate each review word with its POS. POS indicates whether the word is a noun, an adjective and so on. We assume that nouns and noun phrases are the primary candidates for product features, though other constructs (like verb phrases) can be used as well.

2.2. Online review helpfulness prediction

Some literature takes the helpfulness voting ratio as the helpfulness value, formulating the helpfulness prediction problem as a binary classification, a multiple class classification or a regression problem with several different categories of features, such as sentiment features, user reputation or expertise features and information quality-based features. Liu's group utilized reviewer expertise, writing style and timeliness in a linear combination to predict the helpfulness voting ratio. Based on this model, Yu et al. reported their work in predicting the sales of a product. Miao et al. developed a temporal opinion quality aware ranking system to retrieve sentiment information relevant to customers' interest. They built a linear combination model of temporal opinion quality and relevance for the underlying ranking mechanism. Danescu-Niculescu-Mizil conducted a sound analysis of several hypotheses that might influence helpfulness including conformity hypothesis, individual-bias hypothesis, brilliant-but-cruel hypothesis and quality-only straw-man hypothesis. They concluded that helpfulness does not just depend on the content but also “in subtle ways on how the expressed evaluation relates to other evaluations of the same product”.

Different from other helpfulness voting ratio metrics, Liu et al. [17] argued that the helpfulness represented by the helpfulness ratio is not fair due to three kinds of biases: imbalanced vote bias, winner circle bias and early bird bias. Hence, in their work, four human annotators were hired to evaluate helpfulness. They noted that annotators achieved highly consistent results in relation to helpfulness evaluation through a kappa statistic. They modeled the helpfulness evaluation as a multiple class classification problem and employed several categories of features and support vector machines (SVM) algorithm to predict helpfulness. Chen and Tseng proposed a review evaluation framework using information quality. Different from other previous works that only aimed to assess whether product reviews are helpful or not, they classified reviews into five groups: ‘high-quality’, ‘medium-quality’, ‘low-quality’, ‘duplicate’ and ‘spam’. They also utilized several categories of features and SVM to predict the quality of product reviews. Zhang and Tran proposed a helpfulness probabilistic distribution model for helpfulness binary voting (helpful or not helpful) directly from text for online reviews. They utilized expectation-maximization (EM) algorithm to search a distribution that maximizes the helpfulness distribution probability for a given training corpus. The bag-of-words model was used to represent textual content.

2.3. Customer understanding for product design

Customer understanding plays an important role in developing new products. The possibility of a new product's success depends largely on the associated level of customer satisfaction. Based on customer surveys or customer requirement questionnaires, various models were developed in order to maximize customer satisfaction. Tu et al. proposed a cost index structure for the development of an automatic computer-aided cost estimate and control system in mass customization. Such a cost index structure is combined with the generative cost estimate method and the variant cost estimate method to meet customers' needs in a large scale. Mu et al. proposed to bring the Kano Model into quality function deployment (QFD) to quantify customer requirements in an uncertain and vague environment. A fuzzy multi-objective model was then proposed to balance between customer satisfaction and development cost. Kwong et al. proposed a neuro-fuzzy approach to generate a customer satisfaction model. They gave a notebook computer design example to demonstrate that their model is better than a statistical regression approach.

However, all these approaches for customer understanding are subject to the availability of customer survey data, and consumer surveys are well-known to be time-consuming and costly. Although the value of online reviews has been well perceived, it has just started to catch the interest of designers largely due to the technical difficulty being imposed and different knowledge disciplines needed, such as text processing, data mining and information retrieval. Previously, we have proposed an automatic summarization approach based on the analysis of review articles’ internal topic structure to discover and assemble the most important topics that customers have expressed. The
final summary of multiple reviews is then formed based on the topics structure where the topic ranking reveals the importance of different topics. Lee applied a hierarchical two-stage process to supplement traditional methods for assessing rapidly changing user needs based on online reviews [24]. He first used association rules to cluster related attributes and needs into hyper-edges and then searched for hyper-rules relating to hyper-edges in the second stage to track consumers’ requirements. Archak et al. decomposed the online reviews into the characteristics of a product and adapted the methods from the econometrics literature to estimate the weight that customers placed on each individual product feature, the implicit evaluation score that customers assigned to each feature, and how these evaluations would affect revenue [25].

2.4. Quality evaluation of requirement documents in software engineering

In the meantime, some contributions towards customer understanding on requirements documents can be found in the software engineering domain. Previous researchers made efforts on the quality evaluation of software specification documents. Kenett utilized a quantitative approach to assess the quality of software specification documents [26]. Fabbrini evaluated the quality of software requirements by introducing some natural linguistic patterns [27]. They both evaluated the quality of documents using linguistic statistic indicators. But their evaluations have to rely on manual efforts to a large extent. A more specific and automatic evaluation algorithm was proposed by Mu et al. They utilized ten linguistic rules to extract software functional requirements from software requirement specifications [28]. Some machine learning algorithms were also briefly introduced to evaluate whether the requirement specification documents are ambiguous. Hussain proposed a binary classification (ambiguous or unambiguous) using decision tree at both discourse-level and sentence-level [29]. Polpinij suggested to use two classifiers in classifying requirement documents in order to find similar sentences and reduce the ambiguity in the requirement specification [30]. Also, in order to improve the quality and consistency of the written requirement specifications, a requirement specification language, which is similar to human natural language, was proposed [31].

While such efforts also contribute to customer understanding at large, it differs greatly from online review evaluation. First, quality evaluation of requirement documents in software engineering focuses primarily on the effective gathering of explicit design requirements from designated user survey documents which often contain tables, diagrams and questionnaires. In contrast, online reviews are primarily free text largely written by consumers. Their evaluation by designers emphasizes on, for example, its value to suggest designers to certain missing attributes that consumers would wish to have, but designers may not be aware of. Secondly, different from requirement documents in software engineering, online reviews contain a large number of sentences with either strong or weak sentiments. Such sentiments coupled with either existing or potential product attributes attract much more attention from designers. These sentimental expressions contain valuable inputs given by customers and often help designers when they envision new products or improve current models. Thirdly, comparing with a limited number of requirement documents provided in software engineering, designers are often overwhelmed by the sheer number of online reviews, not to mention other technical challenges.

2.5. A brief summary

Through an extensive literature survey from aspects that are deemed relevant, including the efforts on measuring the quality of requirements in software engineering, it helps us to better understand the urgency of our work. Particularly, our review focuses on how the helpfulness of online reviews is currently being defined in the studies primarily in the emerging area of opinion mining and the analysis of online user-generated content. We believe these are the most directly relevant studies.

In the customer-driven design paradigm, many efforts are made towards the understanding and analysis of user inputs and their mapping towards engineering design characteristics, such as quality function deployment (QFD) coupled with fuzzy sets on language processing, models that attempt to maximize user satisfaction and so on. While many efforts have made great advances in different aspects, their successes rely on the quality of user survey data, data that are generated from questionnaires, gathered from focused groups with targeted and structured questions, presented in tables and figures with predefined ranges, specifications, preferences and so on. However, this is enormously different from online product reviews which are entirely free texts written by consumers, in their language with rich sentimental expressions and in a quantity that existing approaches can hardly handle. While opinion mining primarily led by computer scientists also targets at such data, existing efforts from opinion mining stream do not possess a clear focus on how their work is going to help product design and domain users. We intend to suggest reviews that are helpful in the view of designers, even though customer ratings may not be available.

To summarize, we are targeting at a new type of user input that is generically different from those being utilized in existing studies in design, including requirement analysis in software engineering. Such user inputs are valuable to designers, but are not steadily available before the booming of e-commerce and new social media such as blogs and social nets. There is a natural tendency that product end users wish to share their experience of using this product and their preferences online. They wish their experience to be heard by the public, and they also wish to learn from others and better study the characteristics of similar products before they proceed to purchase. Hence, we argue the urgency in studying such crucial sources to better help product design in a great number of constructive ways. The work reported in this paper contributes to the goal that aims to bridge such a notable gap by offering designers and engineers a possibility of processing such a large number of user inputs more effectively and efficiently in a way as helpful as designers and engineers perceive.

3. An exploratory study of helpfulness from designers’ perspective

3.1. Study approach

In order to better understand how helpfulness, as a concept, is being perceived by designers, we have conducted an exploratory study using real-world online reviews. In the previous section, we have presented the fact that, in the current studies, there is a visible gap that comment helpfulness is not currently perceived, defined and evaluated from a domain users’ point of view, like designers and engineers since they are the actual ones who will digest the comments and bring them forward into the design of new models. This motivates us to launch this exploratory study.

Note that the purpose of our study here is not to verify whether the ratings given by designers are right or wrong, but to examine whether the helpfulness evaluation from product designers demonstrates a strong correlation with the online helpfulness voting ratio and attempt to figure out how reviews are deemed as helpful with designers’ evaluations. The task of assigning review helpfulness was carried out by six full-time final year undergraduates in product engineering who are familiar with the review topics, brands and models.
We have randomly chosen 1000 reviews of mobile phones from Amazon. These reviews cover eight different mobile phones. In these 1000 reviews, on the average, there are 300.4 words and 16.5 sentences. However they do not distribute evenly. While the maximal number of words in a single review can reach 3553, there exist a large proportion of short reviews, such as “I absolutely love this phone period. There is nothing else to say. I love it”. Meanwhile, in terms of the number of sentences contained in each review, it also distributed unevenly with the highest number of 222 sentences.

In the study, each student subject has to read all of these 1000 reviews. Different from other research studies, we did not instruct them how to conduct helpfulness evaluation. No annotation guideline was given. The only evaluation concern is to tell whether it is helpful or not helpful towards product design. A five degree helpfulness evaluation was conducted using labels of “0”, “1”, “2”, “3” and “4”. Here, “0” means “least helpful” and “4” means “most helpful”. Finally, some simple policies were explained to the subjects more for coding consistency purpose. Firstly, they were required not to discuss with each other, and the helpfulness evaluation should be judged from the subject’s own perspective based on their knowledge, training and exposure in design engineering. Also, each review should be assigned with the most appropriate helpfulness label.

3.2. Evaluation metrics

In terms of the evaluation, we focus on the comparison between two random variables, i.e., the distance and the correlation between the online voting ratios and the real values (manual ratings). The mean absolute error (denoted by MAE) and the root mean squared error (denoted by RMSE) are adopted to evaluate the distance. They are both frequently used metrics to measure the differences between values presented by a model or an estimator and the values actually observed from the object being modeled or estimated. Also, the Pearson product-moment correlation coefficient (denoted by PMCC) is utilized here to evaluate the correlation between the online voting ratio and actual values. More details are given below.

- Mean absolute error.
  In statistics, the mean absolute error is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. MAE is an average of the absolute errors and is given by

$$\text{MAE}(\text{predict}, \text{real}) = \frac{1}{n} \sum_{i=1}^{n} |\text{predict}_i - \text{real}_i|$$ (1)

$I\text{predict}_i$ is the prediction value and $Ireal_i$ is the true value for example $i$.

- Root mean squared error.
  In statistics, the root mean squared error is also a frequently used measure of the differences between values presented by a model or an estimator and the values actually observed from the thing being modeled or estimated. RMSE of an estimator $predict$ with respect to the estimated parameter $real$ is defined as the square root of the mean squared error. The formula becomes:

$$\text{RMSE}(\text{predict}, \text{real}) = \sqrt{\text{MSE}(\text{predict}, \text{real})}$$

$$= \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\text{predict}_i - \text{real}_i)^2}. $$ (2)

- Pearson product-moment correlation coefficient.
  In statistics, the Pearson product-moment correlation coefficient is a measure of the correlation (linear dependence) between two variables $X$ and $Y$, giving a value between $+1$ and $-1$ inclusive. It is widely used as a measure of the strength of linear dependence between two variables. The formula for the sample PMCC is also available:

$$\text{PMCC}(\text{predict}, \text{real}) = \frac{\sum_{i=1}^{n} \text{predict}_i \cdot \text{real}_i - n \cdot \text{predict} \cdot \text{real}}{n \cdot s_{\text{predict}} \cdot s_{\text{real}}}$$

$$= \frac{\sum_{i=1}^{n} \text{predict}_i \cdot \text{real}_i - \sum_{i=1}^{n} \text{predict}_i \cdot \sum_{i=1}^{n} \text{real}_i}{\sqrt{n \sum_{i=1}^{n} \text{predict}_i^2 - (\sum_{i=1}^{n} \text{predict}_i)^2} \cdot \sqrt{n \sum_{i=1}^{n} \text{real}_i^2 - (\sum_{i=1}^{n} \text{real}_i)^2}}$$ (3)

$predict, real, s_{predict}$ and $s_{real}$ are predicted sample mean, real value mean, predicted sample standard deviation, and real value standard deviation respectively.

3.3. Preliminary results

3.3.1. Number of reviews vs. number of interest voting

Among these 1000 reviews, only 405 reviews received more than five votes originally in Amazon. Fig. 2 provides the statistics results by showing how many reviews receive votes for none, once, and so on.

This fact confirms that only a small part of reviews eventually are voted on by sufficient users. Thus, the online helpfulness voting ratio which is directly utilized as helpfulness evaluation criteria is fundamentally questionable.

3.3.2. Helpfulness ratio vs. averaged helpfulness rating

Due to the helpfulness voting ratio ranging from 0 to 1, we scale the helpfulness voting ratio to the region from $-2$ to $2$ in this evaluation in terms of three metrics. Table 1 shows the results of these values over 1000 reviews.

As seen from Table 1, the averaged student rating presents a weak median correlation with the helpfulness voting ratio. The values of MAE and RMSE account for about 29.5% $(1.178/(2−(−2)))$ and 35.3% $(1.411/(2−(−2)))$ of the scale region respectively. This phenomenon confirms the previous assumption that designers’ helpfulness rating might not present a strong correlation with the
online helpfulness voting ratio and there might be a significant or unacceptable error between them.

3.3.3. Student helpfulness rating

This experiment compared the helpfulness voting ratio with all six students’ ratings. Likewise, the helpfulness voting ratio is also scaled to the region from −2 to 2 and they are also evaluated in terms of three metrics. Fig. 3 depicts the results of these values over 1000 reviews.

Fig. 3 supports the above argument that the online helpfulness voting ratio does not necessarily behave in the same way as designers’ ratings. Even for the smallest MAE and RMSE (student 1), two evaluation metrics still account for about 28.0% (1.12/(2 − (−2))) and 37.3% (1.49/(2 − (−2))) of the scale region respectively. For the PMCC, five of six students only present slightly little higher than 0.25, which shows a weak correlation with the online helpfulness voting ratio.

We also found that the six students presented different preferences on the helpfulness of reviews. One student subject tended to display a strong positive attitude (the average helpfulness evaluation reached 1.03) towards the helpfulness rating on all the 1000 reviews. Another two students tended to give a neutral helpfulness rating (the average helpfulness evaluation are −0.011 and −0.033 respectively.) Another interesting observation is that the perception of helpfulness varies in six subjects’ evaluation. Among these 1000 reviews, only 12 reviews, accounting for slightly higher than 1% of the data set, receive unanimous helpfulness labels from the six subjects. On the other hand, many reviews receive a large standard deviation of helpfulness evaluation. For example, one review was labeled with two “2”s, two “−2”s, one “1” and one “−1”. Six annotators present four different helpfulness evaluations. Another typical example is one review was labeled with five “2”s. This review is also labeled as “−2” by one student. As seen from the result, there exists certain large variance for some reviews. This reveals that different product designers may possess different criteria in evaluating review helpfulness. This triggers us to investigate why a particular review is helpful in the view of designers.

3.4. Why is this review helpful?

First, we intend to know why some reviews receive unanimous helpfulness evaluation. Especially, without giving instructions on helpfulness evaluation, why did the subjects assign the same polarity helpfulness label (i.e., assigning six “−2” labels or six “2” labels to some reviews)? We initiated a questionnaire for all six subjects in order to understand the reasons behind this. According to the questionnaire, some reasons for reviews receiving all “2” labels include “a long review covers his/her preferences”, “mentions many different features”, “points out the like and dislike of the product”, “compares his E71 to Blackberrys”, etc. While reasons for those receiving all “−2” labels include “did not mention anything good or bad about features”, “no information about the performance”, etc.

Second, we are also curious about what has happened to the reviews receiving a large divergence in labeling. So we made another questionnaire to ask the subjects why this was so and their perspectives. Since in this case, every subject’s labeling is different, six different questionnaires were distributed individually. For example, the previous review example receiving two “2”s, two “−2”s, one “1” and one “−1”, subjects who gave label “2” commented that this review “reported that the phone didn’t work properly since the front mask was a little open” and mentioned an “SD card detection problem (requires either OS or Hardware check)”, while those giving label “−2” indicated that this review “only focuses on the problem of the memory card” and “the problem mentioned should not be related to the phone”. For another interesting example with five “2”s and one “−2”, the subject who gave “−2” explained that “customer mainly talks about the usefulness of the phone, the 3rd party applications and not much about the bad sides of the phone”. This follow-up questionnaire further reveals several important perspectives and judgment criteria held by these subjects.

Through this study, we start to understand how designers actually perceive review helpfulness (instead of arbitrarily defining what “helpfulness” should be which happens in existing works). The next task is to find a way to model it, suggest a solution to test the modeling and validate it. This unique exploration serves as the foundation when we propose our modeling approach detailed in Section 4.

4. Technical approach

4.1. Overview of the proposed helpfulness prediction approach

Fig. 4 gives an overview of the proposed helpfulness prediction approach which includes two phases. Assuming that the averaged helpfulness rating by designers is the golden criterion, phase 1 intends to automatically learn and discover a review’s evaluation model from online review content.

In phase 1, based on our understanding of how designers perceive helpfulness, we propose four categories of intrinsic features
Fig. 4. The approach overview.

(i.e., linguistic features, product features, features extracted based on information quality and features extracted using information theory) to support the modeling of helpfulness where the identification and extraction of such features are independent of user ratings and other forms of external knowledge, rules, and dictionaries. The four categories of features are generated entirely from the review contents. We call them intrinsic features. From here and based on the example ratings provided, we propose to search a mapping function between these four categories of features as inputs and the example ratings as corresponding outputs through regression as the first attempt in searching an appropriate algorithm for the learning of a possible underlying model. This concludes the work in phase I.

Since these features are made entirely from the review content only, we venture to explore whether the domain features, one of the four categories of features that we have proposed, possess a strong correlation with the helpfulness perceived. Phase II begins from conducting feature analysis and selection to evaluate whether the categories of domain dependent features will influence the accuracy of helpfulness prediction significantly. We apply the regression model learned from a specific product topic in phase I to predict the helpfulness of unrated reviews, with or without the domain features, on other different product topics. The conjecture here is that if the domain features possess a strong correlation with the helpfulness being perceived, or in other words, without the presence of domain features, the predicted helpfulness will suffer a significant loss in correlation with user ratings. Then the modeling of helpfulness cannot be confidently migrated to another product topic or domain without a re-learning of the helpfulness model using its corresponding user ratings. These serve as the essential research content in phase II.

Therefore, in phase II, we focus on a new and challenging question on helpfulness migration. With a limited number of manually rated reviews which concern a specific product, we intend to explore whether it is possible to model its helpfulness with the intrinsic features that are entirely derived from the review texts alone, and whether such a model is generic enough to be migrated to other products where their manually rated reviews may not be available (implying the absence of new domain dependent features). This is actually the primary reason why our experimental study has been divided into two phases.

4.2 Why are four categories of features chosen?

According to the exploratory study, these students represent their concerns about the helpfulness of online reviews. Some subjects expect that they can learn more useful information from longer product online reviews, primarily indicated by its number of words and its number of sentences. Also, some customers share their comments by telling others their preferences or complaints on this particular product. It is hence valuable for product designers to enquire the reasons behind such sentiments which are mainly expressed using adjective words or adjective plus adverb phrases. Meanwhile, subjects also indicate that they might lose their interest to read and attempt to understand online reviews if there are many grammar errors (number of grammar errors), wrong spellings, and if there are many exceptionally long sentences (average number of words per sentence). This is the reason for linguistic features.

In the meantime, our exploratory study on helpfulness also enlightens us that some product designers focus on whether key product features have been mentioned. For example, a few subjects had given their lists of product features. Each list provides the most important product features the customers have talked about, and such product features are considered crucial information carriers when designers are conceiving a new model. We conjecture that the appearance of some particular product feature might largely influence the helpfulness evaluation. Thus we propose product features as one primary category of features in the model building. According to the two questionnaires, some subjects replied that “this review mentions many product features”, while some argue that “many reviews shared the features he/she likes and
dislikes”. Aggregating all these arguments, as a matter of fact, in a higher and abstract level, we found that these subjects referred to information quality (IQ) in different aspects. For example, the first argument actually mentioned the information coverage and the second points to the information accuracy. All these have inspired us to explore the possibility of extracting features from various perspectives of information quality.

Quite a few subjects, in the exploratory study, gave their most unhelpful labels to reviews with “no (concrete) information about user experience of the product concerned”. It has come to our attention that in this case information can be presented differently from the user perspective in the review. Product feature wise, for example, the sentiment of a product feature that deviates from the majority of sentiment provided in reviews will actually greatly influence designers’ understanding, since it is often associated with more details about why a different or an opposite sentiment is given. Also, a review tends to be regarded as a helpful one if it has mentioned both pros and cons of a product. In the investigation, two subjects explicitly stressed that the appearance of “both pros and cons” is an important factor for helpfulness evaluation. This phenomenon is often referred to as the divergence of sentiments and so it appears in our modeling. In addition, a review stands a higher chance of being rated as helpful if it has expressed a strong and sharp viewpoint towards certain product features with persuasive reasons. It sounds slightly similar to the first observation except the sentiment expressed in this case may or may not align with the rest of the consumers. We propose to interpret such observations using information theory.

4.3. Modeling of helpfulness

In this work, part of our ambition is to explore whether the concept of helpfulness of online product reviews, being perceived from a domain user’s perspective, can be modeled using features that are completely identified, extracted, or transformed from the online view texts. We call them intrinsic features. These features are identified without the assistance of any domain knowledge, like product structure, domain, product ontology, knowledge rules, e-dictionary or thesaurus, and so on. Intrinsic features are entirely derived from the contents of online reviews and serve mainly for the purpose of helpfulness modeling. In the following subsections, we explain how we define four primary categories of intrinsic features based on our understanding from the exploratory study.

4.3.1. Linguistic features extraction

According to the statement in Section 4.2, we employ linguistic features to model the helpfulness of online reviews. Table 2 gives the details of linguistic features that we have proposed.

Several prevailing natural language processing techniques, like part-of-speech (POS) tagging and word sense disambiguation (WSD), are employed in this step. POS tagging is the process of marking up the words in a text as corresponding to a particular part of speech like noun, verb, pronoun, preposition, adverb, adjective or other lexical class marker to each word, based on both its definition, as well as its context, i.e., the relationship with adjacent and related words in a phrase, sentence or paragraph. Also Language Tool,\(^1\) an open source language and grammar checker for English, is utilized to check the number of grammar errors.

4.3.2. Product features extraction

Existing techniques utilize POS tagging for extracting product features. Once POS tagging is performed, linguistic rules or patterns are used to generate feature candidates. For example, “(N) [feature] usage” is a rule, in [11], which matches the segment “battery usage”, to discover feature candidates. “N” is the POS tag that can match any word with that tag, and “usage” is a concrete word which can only match this particular word. After this process, phrases and words with the least possibility of being product features are removed. Clearly, the above involves many interventions and the performance depends on the completeness of the linguistic rules. In our approach, we generate the product feature list based on a document profile (DP) model we previously proposed [12].

The DP model focuses on the discovery of word frequent patterns at the document sentence level which is particularly important for short texts. It requires two parameters as its input, namely, the word gap \(g\) and support value \(s\). The output of the DP model is a list of words and their corresponding frequencies, marked \(\langle w_i, v_i \rangle\), where \(w_i\) denotes the word and \(v_i\) denotes its occurrence value. The DP model is concerned with how to capture single words and word sequences which often bear semantic meaning at the lexical level to represent documents. It extends the basic idea of using point wise mutual information (PMI) to measure the strength of semantic association based on the terms and their co-occurrences. A simple metric called averaged PMI is used to measure the average strength of the semantic association among a set of features.

After generating a list of words from the DP model, we obtain some words with the highest frequency. Notice that product features are often expressed or depicted as nouns or noun phrases. Accordingly, we assume that nouns and noun phrases with the highest frequency are considered product feature candidates though other constructs (like verb phrases) can be used as well. As expected, there still exists some noise, e.g., “phone”, “nokia”, etc. if we regard all these candidates as product features in this process. We need to remove some pre-defined context words with the help of domain knowledge from designers. The DP model also lists words with low frequency, which are actually not product features. We need to prune words with low frequencies that might affect the result. In our experiment, we prune this word list based on the natural distribution of noun words or noun phrases frequency in the data set rather than cutting the word list with some random cut-off ratio. After these candidate selection processes, these noun words or noun phrases left are considered product features. The occurrence frequencies of each extracted product features in product reviews are used as a feature for helpfulness prediction.

4.3.3. Features extraction based on information quality

Several researchers have attempted to identify the possible IQ dimensions that can be used to measure IQ. In this paper, we evaluate review quality primarily based on five aspects: information accuracy, information timeliness, information comparability, information coverage and information relevance. Table 3 lists the details of features extracted based on IQ.

As can be seen from Table 3, there are two tasks, identifying whether a sentence is a subjective one or an objective one for

---

\(^1\) http://www.languagetool.org/.
Table 3

<table>
<thead>
<tr>
<th>IQ aspects</th>
<th>Feature alias</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information accuracy</td>
<td>IQ–NSS</td>
<td># of subjective sentences</td>
</tr>
<tr>
<td>Information timeliness</td>
<td>IQ–TMS</td>
<td># of total elapsed days</td>
</tr>
<tr>
<td>Information comparability</td>
<td>IQ–NRP</td>
<td># of referred products</td>
</tr>
<tr>
<td>Information coverage</td>
<td>IQ–NP</td>
<td># of product features</td>
</tr>
<tr>
<td>Information relevance</td>
<td>IQ–NSPF</td>
<td># of sentences referring to product features</td>
</tr>
<tr>
<td></td>
<td>IQ–RPPR</td>
<td># of product features/# of sentences referring to product features</td>
</tr>
<tr>
<td></td>
<td>IQ–RPFS</td>
<td># of product features/# of sentences</td>
</tr>
<tr>
<td></td>
<td>IQ–RRS</td>
<td># of sentences referring to product features/# of sentences</td>
</tr>
</tbody>
</table>

IQ–NSS and IQ–NOS, and identifying whether a product model name occurs in a review for IQ–NRP, that we need to complete to calculate these IQ dimensions quantitatively. Notice that we still employ the techniques for product feature extraction explained in the previous section for calculating IQ–NPF, IQ–NSPF, IQ–RPPR, IQ–RPFS, and IQ–RRS.

- Identification of sentence sentiment orientation.

Before identifying the sentence sentiment orientation, we need to know whether a sentence contains adjectival words reflecting the author’s sentiment because people tend to express their sentiment with adjectives (sometimes called opinion words). Also, people like to use verbs to express their opinions (e.g., love or dislike). We only consider adjectives as opinion words in this paper. Here we generate two sets of adjectival words, the positive word set and the negative word set. The semantic orientation of an adjectival word indicates the direction that the word deviates from the norm. Adjectives that express a desirable sentiment state (e.g., excellent or perfect) have a positive orientation, while those expressing undesirable states have a negative orientation (e.g., difficult or bad). Notice that the synonyms of an opinion word share a similar sentiment orientation and the antonyms possess an opposite sentiment orientation. We expand the seed word list by WordNet\(^2\) which is a lexical database for the English language, in which English words are grouped into sets of synonyms (called synsets), and it provides short, general definitions, and records the various semantic relations between these synonym sets. Thus, a word is added to the original positive word list if it is a synonym of a positive adjectival word. Similarly, it is added to the original negative word list if it is an antonym of a positive word.

Now we start to judge the sentiment orientation of each sentence. In this work, we utilize the method proposed in [3] to predict the orientation of opinion sentences. We utilize this simple scheme of counting the number of opinion word occurrences to judge the orientation of a sentence. If positive words are greater in number than negative words, the sentence is regarded as positive and vice versa. Also we need to consider the occurrence of the negative modifier word “not”. We flip the sentence sentiment orientation if the word “not” appears.

- Identification of product model name.

We notice that each product \(p_i\) possesses its own product title \(p_i(t)\). Also, each product has an exclusive product model name. For example, “E71” is the unique product model name to represent the “Nokia E71 x Phone”. In our observation, this unique product model name is composed of a combination of digital numbers and characters with “−” or “/”. Based on the discussions above, the algorithm of recognizing a product model name consists of three steps as follows:

**Step 1: Data preparation and pruning.**

This step contains two tasks. Firstly, split the product title \(p_i(t)\) into a string array containing individual words. Secondly, remove those words (e.g., “Nokia”, “with”, etc.) which cannot possibly be a product model name of a product from the string array.

**Step 2: Inverse index generating.**

This step establishes the inverse index for each possible product model name. This inverse index for possible product model name includes two parts: the individual word generated in step 1 and a linked list which contains product identification. As a unique nature for a product model name, a word should be the product model name if the length of the linked list for this word contains only one product identification in this index.

**Step 3: Candidate pruning and model name generating.**

In this step, we aim to prune the candidate model names with the least possibility of being a product model name. Also, we have to keep these words if we cannot decide whether it is a product name. For example, the title of a phone is “Blackberry Storm2 9550 Phone”. It is difficult to guess whether “Storm2” or “9550” is the product model name, so we have to keep both words. After that, we obtain the candidate product model name list. Given the product model name, we utilize an open source project Lucene, a high-performance, full-featured text search engine library with many features including fast indexing, ranked searching, extension APIs, etc. to count the product model name occurrence number for each review.

4.3.4. Features extraction using information theory

In this subsection, information theory is introduced here to estimate the information gained for these three heuristic rules to extract features for helpfulness prediction. Table 4 lists the features extracted using information theory.

Before we calculate the value for the three aspects quantitatively, we need to judge the sentiment for a product feature presented in a review. In this work, we improve the method proposed by Ding et al., where they predicted the sentiment for a product feature by taking the co-occurrence of a product feature and a sentiment word into consideration [32]. Also, they would flip the sentiment of a product feature once a negative word occurs. In this paper, we supply a threshold for the sentiment value of a product feature to further improve this approach. If sentiment value is greater than a certain threshold, we consider that sentence expresses a positive sentiment on the product feature. Similarly, we consider this sentence expresses a negative sentiment on the product feature if sentiment value is smaller than a certain threshold. Otherwise, the sentence expresses a neutral opinion.

- The self-information of product features.

Different reviews present different sentiments for a product feature, and intuitively, this provides different helpfulness information values for designers. Accordingly, for a product feature \(f_j\) extracted as elaborated in the previous section, given a kind of sentiment (positive, negative or neutral), the probability of this product feature for this kind of sentiment, \(\text{prob}(f_j, \text{sentiment})\), in

2 http://wordnet.princeton.edu/.
a dataset is evaluated as:

\[
\text{prob}(f_i, \text{sentiment}) = \frac{\text{num Of sentence}(f_i, \text{sentiment})}{\text{num Of sentence}(f_i)}.
\]  

\[
\text{num Of sentence}(f_i, \text{sentiment}) = \text{number of sentences which express a certain sentiment orientation towards } f_i \text{ and } \text{num Of sentence}(f_i)
\]

According to information theory, given a product feature \( f_i \) and the corresponding sentiment, the self-information gained, SI\((f_i, \text{sentiment})\), is calculated as:

\[
\text{SI}(f_i, \text{sentiment}) = -\text{prob}(f_i, \text{sentiment}) \ast \log(\text{prob}).
\]

Due to the fact that different product features might occur in one review, the total self-information for a review SI\((\text{review})\), is calculated as:

\[
\text{SI}(\text{review}) = \sum \text{SI}(f_i, \text{sentiment}),
\]

We use DS\((f_i)\), for product feature \( f_i \), and SI\((f_i, \text{sentiment})\), for three different sentiments (positive, negative and neutral) and it can be expressed as:

\[
\text{DS}(f_i) = \sum_{\text{sentiment}} \text{SI}(f_i, \text{sentiment}).
\]

Also, due to the fact that different product features might occur in one review, the divergence information, DS\((\text{review})\), for \( f_i \), can be calculated as the sum of divergence information on different product features and it can be expressed as:

\[
\text{DS}(\text{review}) = \sum \text{DS}(f_i),
\]

\( f_i \) denotes the \( j \)th feature in \( \text{review} \). We use DS\((\text{review})\) to estimate the information gained in a different sentiment for a product feature occurring in \( \text{review} \).

- The strength of sentiment sentences.

Review readers tend to prefer the kinds of reviews that express a strong viewpoint towards some product features. Clearly, the strongest sentiment for a product feature might be positive, negative or neutral. We calculate the strength of sentiment, SS\((f_i)\), for product feature \( f_i \), as the maximum of self-information for different sentiments and it can be expressed as:

\[
\text{SS}(f_i) = \text{max}(\text{SS}(f_i, \text{positive}), \text{SS}(f_i, \text{negative}), \text{SS}(f_i, \text{neutral})).
\]

Accordingly, the sentiment strength for \( \text{review} \), SS\((\text{review})\), which mentions different product features, is calculated as:

\[
\text{SS}(\text{review}) = \sum \text{SS}(f_i).
\]

We use SS\((\text{review})\) to estimate the information gained for the designer if the review expresses a strong and sharp viewpoint towards some product features.

4.4. Algorithm description

4.4.1. Algorithms in phase I

As expressed in Section 4.1, we apply the regression model to predict the helpfulness of reviews. We initialized the regression procedure by introducing the bootstrap aggregating algorithm combined with a fast decision tree learner. The fast decision tree learner builds a decision/regression tree using information gain as the splitting criterion, and prunes it using reduced-error pruning. The bootstrap aggregating algorithm is a machine learning ensemble meta-algorithm to improve the classification and regression models in terms of stability and classification accuracy [33]. Given a standard training set \( D \) of size \( N \), the bootstrap aggregating algorithm generates \( M \) new training sets \( D_i \) each of size \( N' \geq N \), by sampling examples from \( D \) uniformly and with replacement. By sampling with replacement, it is likely that some examples will be repeated in each \( D_i \). This kind of sample is known as a bootstrap sample. The \( M \) models are fitted using the above \( M \) bootstrap samples and combined by averaging the output for regression or voting for classification.

Also note that we had actually conducted a series of pre-experiments using other prevailing algorithms like multilayer perceptron neural network (MLP), simple linear regression (SimpleLinear), and sequential minimal optimization for training a support vector regression (SMOreg). They are all popular machine learning algorithms often being utilized in data regression or model complex relationships between various inputs and outputs. MLP is a mathematical model that is enlightened by the structure of biological neural networks. SimpleLinear is the least square estimator of a linear regression model with a single predictor variable. It fits a straight line through the set of \( n \) points in a way that it makes the sum of squared residuals of the model as small as possible. SMOreg globally replaces all missing values and transforms nominal attributes into binary ones.

We utilized a tenfold cross-validation method for testing and reported our results based on the averaged rating of 1000 reviews. Cross-validation is a technique for assessing how the results of an analysis will generalize to an independent data set [34]. It is mainly used to estimate the accuracy of a predictive model in practice. In tenfold cross-validation, the data is randomly partitioned into ten sets of subsamples. A single set of subsamples is retained for testing the model and the remaining sets are used as training data. The cross-validation process is then repeated ten times, with each of the ten subsamples used exactly once as the testing data. We found that the selected algorithm, i.e., bootstrap aggregating plus fast decision tree, performed better than other algorithms.

4.4.2. Algorithms in phase II

In phase II, we had discussed about the possibility of migrating the model discovered to different product topics which actually targets the impact delivered by certain features. Particularly, we chose different feature analysis and feature selection algorithms to identify the most informative and effective features. These feature selection algorithms are:

(1) Principal component analysis (PCA) based feature selection schemes.

Here we apply principal component analysis on three variants of the feature matrix. They include the original feature matrix normalized by its standard deviation, the eigenvalues and eigenvectors of the correlation matrix, and the eigenvalues and eigenvectors of the covariance matrix.

PCA is one well known data preprocessing which is often used as a feature selection and analysis technique. It captures linear
dependencies among attributes of a data set and compresses the attribute space by identifying the strongest patterns in the data, i.e., the attribute space is reduced by the smallest possible amount of information about the original data. Janecek's group conduct PCA on three different variants of the feature matrix to evaluate the classification performance achieved with various machine learning algorithms [35]. They first performed a mean shift of all features such that the mean for each feature becomes 0. Based on this first preprocessing step, they compared the performance of using PCA on the above three different variants of the feature matrix. After the corresponding eigenvalues and eigenvectors are obtained in PCA, features are ranked according to the correlation coefficients of the first principal component and the top ranked features are chosen as the selected features. Likewise, we also applied these PCA based feature selection schemes to evaluate the importance of different features.

(2) Feature-instance similarity based feature selection schemes.

Intuitively, one feature would be regarded as more important if it has the maximum similarity with the instance target. Thus, we introduce some similarity based feature selection schemes. The feature selection schemes in this category calculate the similarity between each feature and the instance target values. The features are then ranked according to the similarity. Three popular metrics are utilized: cosine similarity, Jaccard similarity and matching similarity.

Notice that Jaccard similarity and matching similarity metrics are used for operating on sample sets. Before applying these metrics, the normalized values of each feature and instance target of a matrix are projected to a fixed number of groups. After that, these metrics are applied on the transformed matrix. Features are ranked accordingly and choose the top ranked features as the selected features.

(3) Mutual information based feature selection scheme.

Mutual information is a criterion commonly used in statistical language modeling of word associations and related applications. It is a famous feature selection method, which estimates the mutual information between features and target values, and it was widely used in this field for years. Yang et al. [36] found that mutual information is not an effective feature selection method for text categorization. But no one talked about whether mutual information is an effective method for customer reviews' helpfulness evaluation. Inspired by this thought, the mutual information based feature selection scheme is introduced.

5. Experimental study and discussion

5.1. Experiment setup

In our experimental study, we still used the former 1000 phone reviews collected from Amazon. Since an experimental approach involving two different phases is proposed, we need to evaluate the performance separately. In order to evaluate the feasibility for phase I, the averaged helpfulness value of six students was utilized as the helpfulness value of online reviews. In total, 83 features (including 6 linguistic features, 65 product features, 9 features using information quality and 3 features using information theory) were extracted to predict the helpfulness values. We utilized the bootstrap aggregating algorithm (-P 100 -S 1 -I 10) combined with a fast decision tree learner (-M 2 -V 0.001 -N 3 -S 1 -L 1). The other comparable algorithms, like MLP, SimpleLinear, and SM0reg, were also tested, and we found that the selected algorithm, i.e., bootstrap aggregating plus fast decision tree, performed better than other algorithms. We adopted a tenfold cross-validation method for testing and reported our results based on the average of 1000 repeated experiments.

### Table 5

<table>
<thead>
<tr>
<th></th>
<th>MAE</th>
<th>RMSE</th>
<th>PMCC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.599</td>
<td>0.487</td>
<td>0.795</td>
</tr>
</tbody>
</table>

In phase II, we intend to explore whether this modeling approach is generic enough to be migrated to other products where the manually rated reviews may not be available. Various feature analysis and feature selection algorithms are testified in this phase to examine the conjecture of this specific problem. These feature selection algorithms are: (1) three PCA based feature selection schemes: PCA on the original feature-instance matrix (denoted by PCA), PCA on the correlation matrix of the original feature-instance matrix (denoted by PCACorr), and PCA on the covariance matrix of the original feature-instance matrix (denoted by PCACov), (2) three feature selection schemes based on different feature-instance similarity, i.e., cosine similarity, Jaccard similarity and matching similarity, and (3) mutual information based feature selection scheme.

As for the counterpart review sets under different topics, 904 digital camera reviews and 1026 shaver reviews were randomly chosen from Amazon. Due to the constraint of the budget allocated to pay subjects for their evaluation, in phase II, we could only assign two subjects for each set of reviews. However, according to our previous study [37], we found that "combining two best operators' results is able to achieve close-to-best results" in corpus building. Thus, taking the average of their helpfulness values given as the golden criteria, we compared them with those predicted helpfulness values from phase I which were actually generated from mobile phones—a different category of product. Through this, we evaluated the robustness of the regression algorithm proposed, and examined our conjecture on open-domain helpfulness modeling if these helpfulness values appeared to be highly correlated.

All our programs were implemented and tested in Java 1.6 and Weka 3.6.1 on a dual core 2.40 GHz PC with 4 GB memory. Fig. 5 illustrates the interface of using 1000 mobile phone dataset.

5.2. Results and discussion

5.2.1. Phase I: helpfulness prediction with domain features

Table 5 demonstrates the result of predicted helpfulness value compared with student rating. As can be seen from Table 5, the average predicted helpfulness value indicates a strong PMCC, and small MAE and RMSE, 14.9% and 12.1% of the scale region respectively, with students' rating. It reflects that our proposed model performs much better than a simple helpfulness voting ratio as illustrated in Table 5 and demonstrates that our proposed model is better in interpreting the helpfulness from the designer's viewpoint.

For the algorithm selection of phase I, we did experiments with other algorithms, including a multilayer perception neural network algorithm (MLP), simple linear regression algorithm (SimpleLinear), sequential minimal optimization algorithm for training a support vector regression (SM0reg) and fast decision tree algorithm (REPTree). As for algorithms, all these four algorithms are popular machine learning algorithms often being utilized in data regression to model complex relationships between various inputs and outputs. MLP is a mathematical model that is enlightened by the structure of biological neural networks. SimpleLinear is the least square estimator of a linear regression

---

model with a single predictor variable. It fits a straight line through the set of n points in a way that it makes the sum of squared residuals of the model as small as possible. SMOreg globally replaces all missing values and transforms nominal attributes into binary ones. REPTree builds a decision or regression tree model using information gain reduction. Fig. 6 compares the performance among these different algorithms.

As can be seen from Fig. 6, compared with the other four algorithms, the selected algorithm, Bayesian additive regression kernels statistics model with isotonic regression, performs better in all of the three evaluation metrics (higher PMCC, lower MAE and lower RMSE). Also, the prediction result achieves good performance compared with both the phase I results and the real averaged helpfulness rating.

5.2.2. Phase II: helpfulness prediction without domain features

Fig. 7 compares the reviews’ helpfulness prediction performance of eight feature selection schemes. We utilize all features in these schemes. Mutual information and PCA perform best in all three datasets. For these two schemes, the predicted values reach small distances. Hence, we choose them as our schemes.

Fig. 8 illustrates that PMCC increases (MAE and RMSE decreases) with the number of selected features increasing and the curve reaches relative stability if the number equals nine or ten for both feature selection metrics. These mean that these nine or ten important extracted features have the most influential impact on reviews’ helpfulness evaluation. Table 6 lists these features for both schemes respectively. As can be seen from these features, all of them come from three categories of domain independent features.

As mentioned, in order to further verify the three categories of domain independent features can be applied successfully to predict the online review helpfulness in other domains without losing the prediction accuracy, we need to evaluate whether the predicted helpfulness of online customer reviews from phase I demonstrates a strong correlation with the averaged helpfulness
evaluation rating from design personnel, which will demonstrate that the online customer review helpfulness prediction is not affected by the categories of domain dependent features.

We also evaluate the performance in terms of other different products. The experiment employs 904 digital camera reviews and 1026 shaver reviews to test the previous conjecture. Similarly, another four students (two for digital camera reviews and two for shaver reviews) were invited to label the review's helpfulness according to the same helpfulness labeling instruction explained in Section 3.1. The averaged helpfulness scores from the two students are used as the helpfulness value of online reviews. Notice that only three categories of domain independent features (including 6 linguistic features, 9 features using information quality and 3 features using information theory) are utilized for these two datasets respectively in phase II. Also, we still choose the bootstrap aggregating algorithm combined with a fast decision tree learner as the selected algorithm.

Fig. 9 illustrates the prediction performance by three categories of domain independent features on these two datasets. The predicted helpfulness values on these reviews demonstrate a strong correlation with subjects' helpfulness labeling. Also, the selected algorithm outperforms the other four benchmarking algorithms. The results demonstrate the possibility of modeling online product review helpfulness in the view of designers as an open-domain model.

5.2.3 Feature analysis

Through our user study when the subjects were asked to rate the helpfulness of reviews provided, we had investigated the key question—what are the contributing factors that make the
subjects perceive the comments as helpful. For example, some mentioned that “it has many different features”, “it points out the like and dislike of the product”, etc. The underlying philosophy is that the presence of such features will give an aggregate sense of how useful a particular review would be. In other words, if some features are missing, it implies that the review may not be perceived as useful as others when for example all features are present in the view of designers. At the technical level, these features are weighted and represented in the format of a Weka data file. For the feature vector formation, “0” denotes the absence of the corresponding feature and a non-zero value denotes the feature’s relative importance. Also note since we have features formed based on linguistic information and information quality wise, the feature vector will never be a zero vector. As for feature analysis, we evaluate the importance of different features. As suggested from Table 6, we have found that the absence of some features, e.g., product features, does not influence the helpfulness prediction significantly and different categories of features possess different impacts with respect to the helpfulness value predicted.

From Table 6, it is observed that some linguistic features (L-NW, L-NADJ, L-NADV, and L-NS), have significant influence on helpfulness evaluation. The results echo with a simple helpfulness evaluation assumption that “a longer review tends to be more helpful”. It also infers that a longer review tends to be written by an experienced customer, no matter whether he or she prefers the product or not, and these kinds of reviews tend to be perceived as more than short reviews.

Still, it is noticed that some features (IQ–NSPF, IQ–NOS, IQ–NPF and IQ–NSS) extracted using IQ, also have important impacts on helpfulness evaluation. IQ–NSPF and IQ–NPF confirm the reason, “this review mentions many product features”, is important for helpfulness evaluation. In addition, features IQ–NOS and IQ–NSS reflect the information accuracy for reviews’ helpfulness quality evaluation. This implies that information accuracy for reviews plays an important role in helpfulness evaluation.

IT–SI is another important factor caught our attention when evaluating reviews’ helpfulness. This illustrates that reviews containing a different sentiment that deviates from the majority sentiment tend to be helpful because they might provide further details about this sentiment and make further explanations about the excuses. These would remind product designers for their follow-up work.

Another observation is that a helpful product review may not necessarily compare with many other comparable products (IQ–NRP), which also slightly contradicts with “a helpful review would mention many other different products”. This leads us to believe that there are not many online customer reviews mentioning many different products, a helpful review may only mention one or two similar products for preference comparison, and the reviews’ helpfulness will not always improve much when the number of products mentioned increases.

The feature analysis, using mutual information based feature selection scheme and principal component analysis, explore the utilization of features in terms of how well they can predict the helpfulness. We have verified the applicability of some heuristic-based helpfulness evaluation: “Helpful reviews tend to mention many product features trends”, “Helpful reviews tend to be longer”, “Helpful reviews tend to mention both the advantages and disadvantages of products”, etc. We have also noted that features extracted based on information quality possess a greater influence on helpfulness prediction.

6. Conclusions

We have witnessed an increasing popularity in the analysis of online opinions due to their obvious implications with respect to customer understanding and product design. Based on our comprehensive literature reviews and discussions, we argue that we are actually initiating new research questions of analyzing online product reviews and other valuable online information from a domain user’s point of view and exploring how such online reviews can really benefit domain users. We have observed that there exists a visible gap that helpfulness is not perceived, defined and evaluated from the designers’ point of view. We believe that the ultimate goal of opinion analysis should aim to offer professionals like designers a broad view of customer experiences and insights that provide important clues or evidence for designers to better interpret the voice of the customer, and hence, to refine and improve their existing product offerings accordingly.

Since we have been arguing the notable difference in the nature of product reviews and the urgency of pursuing such topics, in this paper, we have particularly focused on how the helpfulness of product reviews is actually being perceived, defined and evaluated by design engineers. Different from existing efforts, we have conducted a user study to investigate it, come up with a possible way to model the concept of helpfulness from designers’ perspective, and have explored the key question on how design engineers actually perceive and rate the helpfulness of online product reviews. It has come to our attention that there exists a notable difference on ratings between designers and consumers. Based on the insights we have gained, we venture to define four crucial feature categories and have proposed a feasible technical approach to model helpfulness and validate it. This makes our efforts distinct from existing studies in opinion mining and others. Through the user study, we have tested and validated our proposal by many prevailing algorithms in regression and classification. At this moment, we found that the novel combination of many known algorithms works pretty well for the modeling problem. Tested over real-world data, it has achieved a small mean absolute error and root mean squared error, demonstrating a strong correlation between our proposed approach and the designer’s ratings. With such an investigation and the promising results achieved, it leads us to believe that our attempt of modeling helpfulness from designers’ perspectives. Furthermore, our study also shows that it is possible to migrate the model discovered to other product topics without a significant loss, when the manually rated reviews of new topics are not steadily available for training. This work has encouraged us to push our research further that all aim at empowering designers with remarkable business intelligence processing capability.

Acknowledgments

The work described in this paper was supported by a GRF grant from the RGC, Hong Kong SAR, China (RGC Ref: 520208), and was
also supported by a research grant from the Hong Kong Polytechnic University, Hong Kong SAR, China (Grant No. G-RPCV).

References


