Identifying Customer Needs from User-Generated Content

by

Artem Timoshenko

Master of Arts, Economics
New Economic School, 2014

Diploma, Computational Mathematics and Cybernetics
Moscow State University, 2013

Submitted to the MIT Sloan School of Management
in Partial Fulfillment of the Requirements for the Degree

of

Master of Science in Management Research

at the

Massachusetts Institute of Technology
February 2017

© 2017 Massachusetts Institute of Technology. All rights reserved.

Signature redacted

Signature redacted

Signature redacted

MIT Sloan School of Management
January 2, 2017

Certified by:

Kirin Professor of Marketing, MIT Sloan School of Management
Thesis Supervisor

Accepted by:

Catherine Tucker
Sloan Distinguished Professor of Management
Professor of Marketing
Chair, MIT Sloan PhD Program
Identifying Customer Needs from User-Generated Content

by

Artem Timoshenko

Submitted to the MIT Sloan School of Management on January 5, 2017
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Management Research

Abstract

Understanding customer needs is an important part of marketing strategy, product development, and marketing research. The explosive growth of user-generated content (UGC) creates an opportunity to enhance industry-standard interview-based approaches for identifying customer needs. However, the traditional manual review approach is neither efficient nor effective when applied to a large UGC corpus because non-informative and repetitive content crowd out information about customer needs.

We identify customer needs from UGC by combining machine learning methods to select content for review with human judgement to formulate customer needs. In particular, we use a convolutional neural network to filter out non-informative content and dense sentence representations to identify sufficiently different sentences for manual review.

An empirical proof-of-concept compares customer needs for oral care products identified from online reviews (UGC) with customer needs identified by a third-party professional consulting firm using industry-standard methods. In this application, UGC identifies additional customer needs, unreachable by the interview-based approach. Our approach improves efficiency of manual review in terms of a number of unique customer needs per unit effort.

Thesis Supervisor: John R. Hauser
Title: Kirin Professor of Marketing, MIT Sloan School of Management
1. Introduction

Understanding customer needs is an important part of marketing strategy, product development, and marketing research. In marketing strategy, knowledge of customer needs helps to segment the market, to identify the strategic resources for differentiation and to make efficient channel management decisions. In product development, firms analyze customer needs to identify new product opportunities (Herrmann et al., 2000), to design new products (Ulrich and Eppinger 2004, Sullivan 1986) and to improve existing products and services (Matzler and Hinterhuber, 1998). In marketing research, customer needs are used to identify the attributes used in conjoint analysis (Orme 2006).

A customer need is an abstract statement describing the benefits, in customer’s own words, that the customer seeks to obtain from a product or service (Griffin et al., 2009). For example, when describing their experience with the mouthwash, the oral care product, a customer might express the need to know easily the amount of mouthwash to use. This customer need can be satisfied by various product attributes (solutions), including ticks on the cap and textual or visual descriptions on the bottle. Compared to the attributes, customer needs are more abstract, as they describe the customer’s benefit associated with the attribute. Established techniques, such as those used in the application of QFD, identify the correspondence between customer needs and product attributes.

Identifying customer needs requires a deep understanding of a customer’s experience, which is traditionally achieved through interviews and focus groups. Many firms conduct experiential interviews and focus groups and manually identify customer needs from transcripts. Alternatively, other firms use heuristics such as managerial judgment or the review of product comparisons. Recent development of social media and explosive growth of user-generated content (UGC) create both demand and opportunities for complementing the interview-based or heuristic approaches with the analysis of previously unavailable data accumulated online.

Using UGC to identify customer needs provides multiple advantages. First, online content is easily and cheaply obtained and is continually updated. Creating a panel of experienced customers and hiring a professional interviewer is costly. UGC includes information about the
competing products, as well as content related to broader and/or narrower categories. For examples, to identify customer needs for toothbrushes, one could also consider content related to the general category of oral care products or the narrower category of electric toothbrushes. Second, online content enables the firm to clarify customer needs through further analysis of additional UGC. In contrast, it clarification beyond interview transcripts requires new interviews with customers—often an expensive proposition. Finally, a large UGC corpus might represent a broader set of customers allowing more fine-grained segmentation.

To derive customer needs from UGC we must overcome challenges. The firm often does not have sufficient labor resources to review all the available content and is forced to review a small, random sample of the content. In this case, manual review is inefficient because substantial portions of UGC are either non-informative or repetitive. For example, customer might write in a review that a particular retailer is selling a wrong unit. This statement is valid for the review aggregators, but does not reveal information about customer needs. Moreover, if the product has a major defect, customers would discuss it extensively, and information about other customer needs would be crowded out. At the interview-based studies, the content generation is navigated by the interviewer and the firm, so noninformativeness and repetitiveness are less of a problem.

In our study, we design an approach for identifying customer needs from UGC and investigate whether the analysis of UGC complements the traditional interview-based approach. Our approach is a machine-learning hybrid. We use machine learning to address the problems of noninformativeness and repetitiveness by selecting the content for manual review. We draw on recent research in deep learning, in particular, convolutional neural networks (CNN, Kim, 2014) and dense word and sentence representations (Mikolov et al., 2013; Socher et al., 2013). The CNN automatically filters out non-informative content. We use pre-trained dense word representations to represent sentences in a low-dimensional real vector space. Based on the created sentence representations, we sample a diverse set of sentences for manual review. Manual review remains necessary because of the complex nature of customer needs.

To evaluate our approach, we conduct an empirical proof-of-concept study using customer reviews for oral care products from Amazon. We compare the proposed approach to the
current industry-standard practice in terms of a number of unique customer needs per unit effort. We also compare customer needs identified from the UGC to the customer needs identified by professionals using traditional experiential interviews.

Our results have both descriptive and methodological implications. We find out that UGC is more efficient and identifies additional customer needs not reachable by the traditional interview-based approach. The novel hybrid approach is practical and readily implemented.

2. Related Research

2.1. New Product Development

Understanding customer needs is an important topic in new product development literature (Ulrich and Eppinger, 2004). Established techniques, such as those used in Quality Function Deployment (Sullivan, 1986) and conjoint analysis (Green and Srinivasan, 1978; Orme, 2006), use customer needs to identify benefits associated with product attributes and to decide which attributes to include in product design.

The standard approach for identifying and analyzing customer needs, the Voice of the Customer (VOC), is interview-based (Alam and Perry, 2002; Griffin and Hauser, 1993). It typically involves three steps. First, multiple experiential interviews or focus groups are conducted. Next, human analysts review transcripts to identify a set of customer needs and remove redundancy (winnowing). The third step uses either affinity groups or clustered co-occurrence matrices to identify a hierarchical structure with three levels corresponding to primary, secondary, and tertiary needs. After customer needs are identified, customers may be asked to identify priorities over the set of customer needs. Identifying customer needs is the most expensive part of the VOC studies. Our paper examines how UGC could be used to complement the VOC results and proposes a method for identifying customer needs from UGC.

2.2. Automated Marketing Research

The analysis of unstructured textual UGC is an important topic in marketing literature. For example, Lee and Bradlow (2011) demonstrate how UGC could be used for mining product attributes and levels from customer reviews data, and identifying strategic positioning. Netzer
et al. (2012) propose an automatic approach for generating market-structure perceptual maps from UGC.

2.3. Deep Learning for Natural Language Processing

We draw on two literatures from natural language processing (NLP) – convolutional neural networks (CNNs) and dense word and sentence representations. CNN is a supervised prediction technique, which combines convolutional and pooling layers to process inputs of various size. In NLP, CNNs demonstrate state-of-the-art performance with minimum tuning in such problems as relation extraction (Nguyen and Grishman, 2015), named entity recognition (Sun et al., 2015) and sentiment analysis (Santos and Gatti, 2014). We train a CNN to eliminate non-informative content from consideration.

The dense word and sentence representation are the low-dimensional real vector mappings, which are trained such that representations of similar words or sentences are nearby at the numerical space. The fundamental idea for training the dense representations in NLP is the Distributional Hypothesis, which states that words that appear in the same context share semantic meaning (Harris, 1954). The use of neural network models allows training high-quality representations that can be applied in downstream NLP applications and models (Lample et al., 2016; Kim, 2013). We use word and sentence representation to reduce repetitiveness of content selected for manual review.

3. Problem Formulation (Design Requirements)

Suppose, a firm has a UGC corpus which is split into $L$ sentences. The firm wants to analyze and derive a list of non-repetitive customer needs. The analysts at the firm can read sentences and infer all needs articulated in the sentences. The firm has capacity to review $N$ sentences, where $N \ll L$.

In current practice analysts select a random set of $N$ sentences for review. If the density of needs per sentence is $k$ where $k$ can be less or greater than 1, the expected number of needs identified in the set of $N$ sentences, would be $kN$. After removing redundancy, the firm
identifies $K_0$ needs where $K_0 < kN$. Typically, $K_0 \ll kN$. Our goal is to design an approach to identify more than $K_0$ unique customer needs from a corpus of length $L$ such that the analysts' effort does not exceed, and hopefully is substantially less than, that required to manually review $N$ randomly-drawn sentences.

There are two important properties of the desired solution. First, completeness of the set of identified customer needs and precision of their formulation have higher priority than complete automation in our setting. Customer needs should be formulated in a way relevant for subsequent analysis by methods such as QFD or conjoint analysis. We seek to be as complete or more complete than existing methods.

Second, we want the solution to be scalable. As the amount of UGC grows, the desired solution should be able to produce an output in a reasonable time and with non-decreasing performance in terms of an expected number of identified unique needs.

4. Methodology

Our approach for identifying customer needs from UGC involves the following three stages.

1. Use a machine learning classification technique, a convolutional neural network (CNN), to filter out non-informative sentences from consideration. We train the CNN on a small set of sentences manually reviewed and labeled 'informative or 'non-informative.'

2. Use pre-trained dense word representations to create sentence vector representations which summarize the content of the informative sentences. Cluster sentence representations and sample from clusters to identify a diverse set of informative sentences.

3. Manually review the selected sentences and identify customer needs using trained industry analysts.

We conduct our analysis at the sentence level because sentences are convenient for both manual review and machine learning.

4.1. Stage 1 – Identifying Informative Content

Identifying informative sentences at the first stage of the analysis is extremely important. Non-informative sentences require that we correct techniques to account for irrelevant content
which complicates the further analysis. The added noise from irrelevant content overwhelms our ability to process relevant content. Focusing manual effort on informative content is more efficient and increases the probability of identifying new customer needs.

To identify informative content, we train a predictive model. We generate a training set by manually reviewing a small set of sentences. Each sentence is labeled ‘informative’, if it helps to identify customer needs, and ‘non-informative’ otherwise. This training set enables us to train, that is, set parameters, for a model which can identify informative sentences in the rest of the corpus. Generating the training set requires human effort, but, overall, this effort enhances the efficiency of the process.

As an illustration, suppose a firm has resources to manually review 5,000 sentences, and only 10% of sentences at the UGC corpus are informative. The current practice is that the firm randomly selects 5,000 sentences for manual review, yielding 500 informative sentences. Suppose instead that the firm could review 1,000 sentences and use them to train a machine learning classifier. For illustrative purposes, assume the classifier has a 100% hit rate and it is applied, without human effort, to a large UGC corpus. All sentences identified by a classifier are informative. In this case, the analysts can sample additional 4,000 sentences from the ones identified by a classifier and review 5,000 sentences total again. 4,100 sentences are informative now – 10% of the initial 1,000 sentences and all of the remaining 4,000 sentences—more than an eightfold increase in efficiency. Even if the classifier is not perfect, the gains can be substantial. If the precision of the classifier is 90%, the firm identifies for review 3,700 informative sentences—a net increase of 3,200 additional informative sentences for the same level of manual effort. This 740% increase in the number of informative sentences significantly increases the chances that the firm will identify new customer needs, and thus, improve the overall quality of the study.

When the number of sentences that can be analyzed manually is a constraint, there is a tradeoff between the size of the training sample and the size of the analyzed corpus. More training makes the classifier more accurate, but a sentences selected for review randomly also increases. Suppose, using 2,000 sentences for training instead of 1,000 sentences increased the hit rate from 90% to 100% at the example above. With 1,000 sentences at the training set,
the firm would identify 3,200 informative sentence, while it would identify 3,700 informative sentences when the training set consists of 1,000 sentences — 100 from the training sample and 3,600 from the sentences identified by a classifier. In this case the firm would be best advised not to increase the training sample from 1,000 to 2,000 sentences.

We evaluate the quality of a classifier using two criteria: precision and recall:

\[
\text{precision} = \frac{TP}{TP + FP}
\]

\[
\text{recall} = \frac{TP}{TP + FN}
\]

where \(TP\), \(FP\) and \(FN\) correspond to a number of identified true positives \((TP)\), false positives \((FP)\), and false negatives \((FN)\) at a holdout set. Precision is the share of informative sentences among the sentences identified as informative by a classifier; recall is a share of informative sentences correctly identified by a classifier. In machine learning, it is common to combine these measures into an \(F_1\) score as the primary measure of interest (Wilson et al., 2005).

\[
F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\]

While efficiency is valuable, we want to be assured that the informative sentences are not limited to identifying customer needs that appear in the training sample. In particular, suppose the training sentences articulate customer needs \(C_1, \ldots, C_{N_0}\). The classifier used in our approach should be able to correctly identify sentences from the remaining corpus where new needs are articulated. That is, the informative sentences from the remaining corpus which contain customer needs that are beyond \(C_1, \ldots, C_{N_0}\).

**Convolutional Neural Network**

We use a deep convolutional neural network (CNN) to identify informative sentences. Figure 1 demonstrates an example of a CNN architecture typical for natural language processing (NLP) applications.
For every word in the dictionary, CNN stores word representations. In the literature, word representations are either vectors with a single "1" corresponding to the index of the word at the vocabulary, also known as one-hot vectors (Johnson and Zhang, 2015) or are low-dimensional real vectors (Collobert et al., 2011). When low-dimensional word representations are used, they can be treated as parameters of the model and calibrated along with other parameters of the model, or they can be pre-trained. Our CNN uses pre-trained 300-dimensional word-vector representations. We provide more detail in §4.2.

Using stored word representations, the CNN represents an input sentence as a concatenation of numeric representations of the words in the sentence.

$$\mathbf{v} = [v_1, ..., v_n] \in \mathbb{R}^{d \times n}$$

where \(d\) is a dimensionality of word representations, and \(n\) is a length of the input sentence. The CNN uses \(\mathbf{v}\) as an input for convolutional, pooling and softmax layers to classify the initial sentence.

Convolutional layers create multiple “feature maps” by applying convolutional operations with varying filters to the vector \(\mathbf{v}\). A filter is a real vector \(w_t \in \mathbb{R}^{d \times h_t}\), where \(h_t\) is a size of the filter. Filters are applied to different parts of the vector \(\mathbf{v}\) to create feature maps:

$$c^{w_t} = [c^{w_t}_1, ..., c^{w_t}_{n-h_t+1}]$$

$$c^{w_t}_i = \sigma(w_t \cdot v_{i:i+h_t-1} + b_t)$$

where \(\sigma(\cdot)\) is a non-linear transformation, e.g., \(\sigma(x) = \max(0, x)\), which is called activation.
function, \( b_t \in \mathbb{R} \) is a bias, and \( v_{i:i+h_t-1} \) is a concatenation of representations of words \( i \) to \( i + h_t - 1 \) in the sentence:

\[
v_{i:i+h_t-1} = [v_i, \ldots, v_{i+h_t-1}]
\]

Size of the filters, \( h_t \), can vary within the convolutional layer. The number of filters and their size are hyperparameters of the model, which are selected before the training and can be tuned at the cross-validation. Filters are parameters which are calibrated at the training. Figure 2 illustrates a process of generating a feature map with a filter of size \( h_t = 2 \).

![Figure 2](image.png)

Figure 2. This scheme shows how the convolutional operation \( \sigma(\cdot) \) with a filter \( w_t \) of size \( h_t = 2 \) generates a feature map \( c^{w_t} \).

If multiple convolutional layers are stacked together, the number of parameters to be calibrated explodes and the CNN becomes computationally infeasible. The same problem arises when the number of feature maps is large. To reduce dimensionality of the output of the convolutional layers, pooling layers are added to the CNN architecture.

Common operators for pooling in NLP applications are max-pooling or \( k \)-pooling (Collobert et al., 2011). Max-pooling receives a feature map as an input and identifies a largest feature. \( k \)-pooling identifies \( k \) largest features at the feature map. The output of the last pooling layer is a vector \( z \) that is a concatenation of the results of pooling operators applied to feature maps:

\[
z_t = \max c^{w_t} = \max[c_1^{w_t}, \ldots, c_{n-n+h+1}^{w_t}]
\]

\[
z = [z_1, \ldots, z_T]
\]

The last layer of the CNN, called softmax layer, uses vector \( z \) to perform the final classification of the sentence. The classification is based on a binary logit model.
Suppose, a sentence can belong to one of two classes: $y = 1$ or $y = 0$. In our study, $y = 1$ if the sentence is informative, and $y = 0$ otherwise. Then, the softmax layer uses vector $z$ to estimate the probability $P(y = 1|z)$.

$$\hat{y}(z) = \hat{p}(y = 1|z) = \frac{1}{1 + e^{-z^T \theta}}$$

where $\theta$ is a vector of parameters.

Multiple convolutional and pooling layers can be combined to achieve better performance (Kalchbrenner et al., 2014). A particular specification depends on the application and can be varied at the development process. In our study, we use a CNN specification from Lei et al. (2015).

The number of parameters at the CNN depends on the architecture of the network. In Figure 1, the parameters of the model include word representations, $v_w \in \mathbb{R}^d$, for all words in the vocabulary, filters, $w \in \mathbb{R}^{d \times h}$, in convolutional layers, and weighting vectors, $\theta$, in the softmax layer. In our study, we use pre-trained word representations and calibrate parameters $w$ and $\theta$ by minimizing the cross-entropy error on the training set of manually labeled sentences:

$$\hat{w}, \hat{\theta} = \text{argmax}_{w, \theta} L(v, w, \theta)$$

$$L(v, w, \theta) = -\frac{1}{N} \sum_{n=1}^{N} [y_n \log \hat{y}_n + (1 - y_n) \log (1 - \hat{y}_n)]$$

where $N$ is a size of training set, $y_n$ are the manually assigned labels, and $\hat{y}_n$ is the prediction of the CNN.

The primary advantage of the CNN approach over traditional machine learning classification techniques, such as support vector machines (SVMs), is automatic feature extraction. Automated feature extraction makes a CNN ideal for use across a broad range of applications.

The performance of an SVM depends significantly on the quality of features. The best features are usually handcrafted specifically to application and context, so feature generation might require a substantial human effort. A CNN uses convolutional layers applied to the
word-vector representations to extract features from the textual data. CNNs demonstrate a comparative performance to such handcrafted SVMs on most benchmarks (Lei et al., 2015; Kim, 2014).

4.2. Stage 2 - Sentence Representations and Sampling

In Stage 2, we resolve the problem of repetitiveness of informative sentences identified in UGC. Repetitiveness is a major problem for manual review. Not only is repetitiveness inefficient, but too much repetitiveness makes the task onerous and boring for human reviewers. We want to minimize redundancy by sampling sufficiently different sentences from the corpus.

Recall that, in §4.1, we used pre-trained low-dimensional vectors to represent words. The vectors that we used were trained such that representations of semantically similar words are close in the vector space. Vectors with this property are called “word embeddings.” We use word embeddings to minimize redundancy.

High-quality word embeddings can be pre-trained on a large text corpus and then applied in downstream NLP tasks (Baroni et al., 2014). In our study, we use 300-dimensional word embeddings pre-trained on the Google News Corpus using a Skip-gram model (Mikolov et al., 2013a). The Skip-gram model is a predictive model, which trains the embeddings of words in the sequence \( w_1, \ldots, w_T \) by maximizing the average log-likelihood of words appearing together (within \( c \) words) in a sequence. That is, we maximize:

\[
\frac{1}{T} \sum_{t=1}^{T} \sum_{j \neq 0}^{|V|} \log p(w_{t+j}|w_t)
\]

\[
p(w_b|w_a) = \frac{\exp(v'(w_b)^T v(w_a))}{\sum_{k=1}^{|V|} \exp(v'(w_k)^T v(w_a))}
\]

where \( V \) is the set of all feasible words (the vocabulary) and \( v(w_i) \) are word embeddings.

Word embeddings so-trained capture semantic information about words and semantic relationship between them (Mikolov et al., 2013b). For example, the word embeddings that we use have properties such as:
\[ v(\text{king}) - v(\text{man}) + v(\text{woman}) \approx v(\text{queen}) \]
\[ v(\text{walking}) - v(\text{swimming}) + v(\text{swam}) \approx v(\text{walked}) \]
\[ v(\text{Paris}) - v(\text{France}) + v(\text{Italy}) \approx v(\text{Rome}) \]

Word embeddings can be used to create representations for sentences. Established methods for creating sentence representations include taking a simple average of word embeddings of words in the sentence (Iyyer et al., 2015), explicitly modeling semantic and syntactic structure of the sentences with neural methods (Tai et al., 2015), or training sentence representations together with word representations at the prediction task (Le and Mikolov, 2014). Because sentence representations obtained through averaging of word embeddings demonstrate similar performance to the methods explicitly modeling semantic and syntactic structure of the sentences (Iyyer et al., 2015) and they are easy to compute, we use averaging for our proof-of-concept application.

With defined sentence representations, our goal of providing sufficiently different sentences for manual review can be viewed as an analogy to sampling sentences from different areas of the numeric space. We use a standard Euclidean distance metric to measure similarity between sentences, and we group sentences into multiple clusters representing similar topics. We choose Ward’s clustering method because it is common in voice of the customer methods, e.g., Griffin and Hauser (1993), and has been perfected over hundreds of applications in the last twenty-five years. Following standard practice, we select the number of clusters by the elbow method.

We randomly sample sentences for manual review from different clusters (Sarkar, 2009). The number of sentences drawn from each cluster is proportional to the size of the cluster to guarantee that sentences for manual review are sampled uniformly from all areas of the sentence representation space.

4.3. Stage 3 – Manually extract customer needs

To date, we have not been able to identify a machine-automated approach to extract customer needs from informative sentences. Extracting customer needs from the informative sentences requires a deep understanding of the customer feedback, which is better done manually. Using
machine learning to identify informative sentences and then trained analysts for the final step appears to be the best combination of automation and manual analysis.

We adopt standard methods to extract customer needs from the sentences selected at Stage 3. Trained analysts, with experience in traditional VOC analyses, manually review sentences and highlight all customer needs that can be inferred from the informative sentences.

Following standard industry procedures, we combine identified customer needs into one set and manually remove duplicate customer needs from the set. The final output is a list of non-repetitive customer needs, which can be compared to the output from a traditional industry-based VOC study.

5. Proof of Concept and Empirical Comparison

We evaluate the efficiency of identifying customer needs from UGC on a set of Amazon reviews for oral care products. First, we verify whether analysis of UGC can complement interview-based studies by identifying additional customer needs. Second, we compare the performance of our approach for identifying customer needs to the performance of the current practice. For the purposes of this paper, we call the results of the interview-based study the “gold standard.”

5.1. Gold Standard

A professional marketing consulting firm with almost thirty years of experience in VOC studies provided us with a hierarchy of customer needs on oral care products. The hierarchy was created for client who used the VOC to develop new products in oral care. Customer needs were identified and organized into a hierarchical structure using traditional interviewing and card-sorting methods (e.g., Griffin and Hauser, 1993). The customer-need hierarchy represents 6 primary needs, 22 secondary needs, and 86 tertiary needs. We selected the oral care product category, because customer needs associated with oral care products are relative broad, and because the number of tertiary needs is not too large as, for example, for autos or smartphones. The oral care category provides sufficient variation and does not overcomplicate the comparisons.
5.2. UGC Data

We consider a corpus of 115,099 Amazon reviews on oral care products spanning the period from 1996 to 2014. We use an unsupervised sentence tokenizer (Kiss and Strunk, 2006) from natural language toolkit (NLTK) to automatically split text into 408,375 sentences.

In order to evaluate the proposed approach, at our request, analysts at the marketing consulting firm reviewed 8,000 randomly-selected sentences of UGC and identified needs articulated at each sentence. The identified needs were assigned with the need IDs corresponding to the tertiary needs at the gold standard. When the identified (tertiary) need was not present at the gold standard, the marketing consulting firm tried to assign it to the existing primary and secondary need groups. If two or more new needs were assigned to the same group, they received the same new need ID, and we cannot differentiate them at the final dataset. The final dataset consists of 8,000 sentences with possibly multiple need IDs assigned to each sentence.

5.3. Descriptive Statistics

Informative sentences constitute 52% of the 8,000 sentences in the Amazon reviews dataset. Amazon reviews are focused on describing products and customer experience and are likely rich in customer needs. We expect that uninformative content will provide a greater challenge in other sources of UGC, such as social media or blogs.

We observe a high degree of repetitiveness in the oral-care corpus. Customers tend to express similar needs in the reviews. In particular, 10% of the most frequent customer needs are articulated in 54% of the informative sentences.

Figure 3 summarizes the relationship between the customer needs identified from the interviews and the customer needs identified from UGC. Even limiting ourselves to the 8,000 sentences for which we have detailed comparative data, UGC identified 74 of 86 tertiary needs identified from directed, experiential interviews. Likely, the remaining twelve needs would have been identified had we used a larger portion of the oral-care corpus.

In product development, practitioners are always seeking “unarticulated customer needs.” Unarticulated customer needs often provide opportunities to surprise and excite customers
and, thus, lead to successful new products. Our analysis of the oral-care UGC corpus identified seven groups of customer needs (need IDs) that were not articulated in experiential interviews. Perhaps such needs might have been articulated had the gold standard been based on more interviews, but new customer needs from UGC provide new opportunities for oral-care products. At minimum, we conclude that UGC complements interview-based methods.

![Diagram showing customer needs identified from interviews and UGC.](image)

**Figure 3.** Customer needs identified from interviews and UGC.

### 5.4. Implementation Details

#### 5.4.1. Preprocessing

We preprocess sentences before applying machine learning techniques. In particular, we eliminate stop-words (e.g., ‘the’ and ‘and’) and non-alphanumeric symbols (e.g., question marks) and transform numbers into number signs and letters to lower case.

Among the 8,000 sentences in the dataset, 1,394 sentences contain either more than ten, or less than three words after preprocessing. We treat such sentences as an artifact of the grammatical or punctuation errors in UGC, which leads to a lower accuracy of the NLTK sentence tokenizer. We drop these sentences and evaluate the machine learning methods using the remaining 6,606 sentences.

#### 5.4.2. Convolutional Neural Network

To train a CNN, we split 6,606 sentences into training, development, and test sets. We calibrate parameters of the model using the training set, tune hyperparameters of the CNN on the development set, and evaluate its performance on the test set. We call the sentences identified as informative by the trained CNN at the test set the “identified set.”

As shown in Figure 4, precision, recall and the F1-score tend to increase as the training set becomes larger. The CNN has a large number of parameters, so it overfits if the size of the
training set is small. For example, when the size of the training set is 500 observations, the CNN classifies almost all sentences as informative. In our study, the calibration of parameters stabilizes when the number of observations in the training set is at least 1,000. We expect the threshold to depend on the particular specification, including the number of layers, and on the particular application, but the fact that training stabilizes after 1,000 observations means that CNNs are likely feasible for a broad range of product categories.

![Figure 4. Accuracy of the CNN depends on the size of the training set](image)

The optimal number of sentences to use in training represents a trade-off. Suppose that the precision of the CNN trained on \( x \) sentences is \( f(x) \), and \( f(0) \) is a share of informative sentences in the random sample of UGC. Then, if the firm is limited in its capacity to review sentences, say to \( N \) sentences, it maximizes the expected number of reviewed informative sentences by choosing \( x \) to maximize

\[
\max_{x} x f(0) + (N - x) f(x)
\]

We want to review fewer sentences for training, but the precision of CNN trained on a small sample decreases. As of this writing, we have not estimated \( f(x) \) and cannot determine the optimal \( x \) for the 6,606 sentences in our coded corpus.

Figure 5 illustrates how often customer needs are articulated in the training set, in the test set, and in the identified set. The CNN correctly identifies informative sentences, even when the
corresponding customer need is rarely articulated at the training set. Moreover, the customer need for using as few products as possible and still having an effective oral care routine is correctly identified by the CNN, but is never articulated in the training set.

![Figure 5. Frequency of customer needs at the training, test, and identified sets.](image)

5.4.3. Sentence Representations

Having identified informative sentences, we now create sentence representations which help to sample content for manual review to span the space of customer needs efficiently. We calculate sentence representations by averaging 300-dimensional word embeddings pre-trained on the Google News corpus (Mikolov et al., 2013). Because it is difficult to visualize a 300-dimensional vector, we use principle components analysis to project sentence representations to two dimensions. This projection is shown in Figure 6 for two primary need groups. Red dots correspond to the sentences which identify customer needs related to the teeth and gum strength. The blue dots correspond to sentences which identify customer needs related to shopping or product choice. Even in the two-dimensional projection, we observe that the two groups can be separated. Such separation suggests that sentence representations capture relevant information about the corresponding customer needs, and they can be used to identify similarity between sentences.
5.5. Results

Current industry practice for identifying customer needs from UGC is based on a manual review of a random sample of the UGC corpus. We consider such random sampling as a baseline approach. In our approach, we randomly sample a subset of sentences to train a CNN, and then, use the CNN and sentence representations to improve subsequent sampling by focusing on a diverse set of informative sentences.

To evaluate our approach, we consider a CNN trained on 5,000 sentences. With 5,000 sentences, the CNN achieves a high performance in terms of F1-score and the held-out 1,606 sentences are sufficient for a preliminary evaluation. This CNN achieves 75.5% precision and 78.4% recall on the test set, and identifies 1,040 sentences as informative.

An exhaustive review of the held-out sample identifies 71 unique customer needs. Suppose, we have resources to manually review only \( N \) sentences, where \( N \ll 1,606 \). For every \( N \), we evaluate content selection approaches in terms of the expected number of identified unique customer needs.
Figure 7. Comparison of content selection methods

Figure 7 summarizes the results of the evaluation. The dashed line corresponds to the baseline of random selection (current practice). The dash-dot line corresponds to using the CNN, but not clustering sentence representations. The dotted line corresponds to using both the CNN and clustering sentence representations to ensure diversity. Each line represents the number of customer needs identified if only that number of sentences were reviewed (250, 500, 750, and 1,000).

In Figure 7, our approach outperforms the baseline in terms of the expected number of identified unique customer needs. The CNN improves efficiency of the process for all $N$. Clustering sentence representations ensures diversity of the content selection and improves efficiency for smaller $N$. The value of diversity decreases as $N$ increases toward 1,000.

Figure 7 can also be read horizontally. To identify 63 customer needs, the current practice requires reviewing 750 sentences. Our approach reaches the same number of customer needs with 600 sentences. Thus, our approach gains 20% improvement in efficiency of manual review. More importantly, our machine-learning approach is a first application whereas the standard approach has been perfected in almost thirty years of application. It is supported by sophisticated training, refined software, and experienced analysts. We expect our proof-of-concept performance to improve dramatically with application.
6. Conclusions

In this paper, we examine how the traditional marketing techniques for identifying customer needs could be complemented by the analysis of UGC. We find out that UGC allows identifying customer needs not reachable by the interview-based approach. We propose an approach for identifying customer needs from UGC which is based on a combination of machine learning and human judgement. The proposed approach involves using deep learning methods to filter out non-informative content and to reduce repetitiveness of the content, but it relies on human judgement at the final stage to formulate customer needs.

Further Work

(1) Our analysis indicates that UGC allows identifying customer needs not reachable by the interview-based approach. On the other hand, some customer needs identified at the interviews are never identified at the large corpus of UGC. Understanding the differences between customer needs uniquely identified in either UGC or interview data will uncover in which cases analysis of UGC provides the most value.

(2) Using deep neural networks for sentence classification and sentence embedding are active areas of research in NLP community. We expect performance of the proposed approach for identifying customer needs to improve significantly with further development of machine learning techniques.

(3) As UGC is continuously updated, the firms want to identify new customer needs on the fly. The proposed approach for identifying customer needs from UGC requires manual effort, so it requires adjustment to allow efficient processing of the stream of data.
Bibliography


