Identifying Customer Needs from User-Generated Content

by

Artem Timoshenko

and

John R. Hauser

February 2017

Artem Timoshenko is a PhD student at the MIT Sloan School of Management, Massachusetts Institute of Technology, E62-584, 77 Massachusetts Avenue, Cambridge, MA 02139, (617) 803-5630, atimoshe@mit.edu.

John R. Hauser is the Kirin Professor of Marketing, MIT Sloan School of Management, Massachusetts Institute of Technology, E62-538, 77 Massachusetts Avenue, Cambridge, MA 02139, (617) 253-2929, hauser@mit.edu.

We thank John Mitchell, Steven Gaskin, Carmel Dibner, Andrea Ruttenberg, Patti Yanes, Kristyn Corrigan and Meaghan Foley for their help and support. We thank Ken Deal and Ewa Nowakowska for suggestions on earlier versions of this paper. This paper has benefited from presentations at the 2016 Sawtooth Software Conference in Park City Utah, the MIT Marketing Group Seminar, and presentations at Applied Marketing Science, Inc. and Cornerstone Research, Inc.

Identifying Customer Needs from User-Generated Content

Abstract

Identifying customer needs is important to marketing strategy, product development, and marketing research. User-generated content (UGC) provides an opportunity to better identify (abstract) customer needs. However, manual review is neither efficient nor effective for large UGC corpora because much content is non-informative and repetitive. We propose a machine-learning approach to select content for efficient review. We use a convolutional neural network to filter out non-informative content and cluster dense sentence embeddings to avoid sampling repetitive content. We further address two key questions: Are customer needs identified in UGC comparable to customer needs identified with standard methods? Do the machine-learning methods improve customer-need identification. Our benchmark is a set of customer needs identified by a professional analysts using standard experiential interviews. Professional analysts coded 12,000 UGC sentences to identify if each sentence contained one or more previously identified customer needs and/or a new customer need. Results: (1) Holding coding effort constant, the UGC corpus contains comparable needs to experiential interviews. (2) Holding overall effort constant, the corpus of UGC is better able to identify customer needs than traditional methods. (3) Machine-learning methods improve efficiency (unique customer needs identified per unit effort).

Keywords: *Voice of the Customer; Machine Learning, User-generated Content; Customer Needs; Online Reviews; Market Research; Text Mining; Deep Learning; Natural Language Processing*

1. Introduction

In marketing strategy, customer needs help segment the market, identify strategic dimensions for differentiation, and make efficient channel management decisions. For example, Park, Jaworski, and MacInnis (1986) describe examples of strategic positioning based on fulfilling customer needs: "attire for the conservative professional (Brooks Brothers)" or "a world apart—let it express your world (Lenox China)." In product development, customer needs identify new product opportunities (Herrmann, Huber, and Braunstein 2000), improve the design of new products (Krishnan and Ulrich 2001; Sullivan 1986; Ulrich and Eppinger 2004), help manage product portfolios (Stone, et al. 2008), and improve existing products and services (Matzler and Hinterhuber 1998). In marketing research, customer needs help to identify the attributes used in conjoint analysis (Orme 2006).

New "excitement" customer needs are particularly important for product development (Kano, et al. 1984; Mikulić and Prebežac 2011). For example, consider the breakthrough laundry detergent, "Attack," developed by the Kao Group. Before Kao's innovation, firms such as Procter & Gamble thought they knew all (primary) customer needs: cleaning, safe and gentle, good for the environment, ready to wear after drying, easy to use, smell fresh and clean, and value. Because most products fulfilled these customer needs, perceived "value" played a major competitive role. Detergents were sold in large "high-value" boxes. Kao recognized that Japanese consumers did not have space for large boxes in their apartments, nor could they transport them easily by foot or bicycle from the store. To fulfill this newly-identified customer need, Kao launched a highly-concentrated detergent in an easy-to-store and easy-to-carry package. Despite a premium price, Attack quickly commanded almost 50% of the Japanese laundry market (Kao Group 2016). American firms soon introduced their own concentrated detergents.

A customer need is an abstract statement describing the benefits, in the customer's own words, that the customer seeks to obtain from a product or service (Brown and Eisenhardt 1995; Griffin, et al., 2009). For example, when describing their experience with mouthwashes, 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. Abstract customer needs are described by precise sentences or phrases rather than “bags of words.” For example, oral care products might be able to “maintain a natural shade of white that doesn’t look phony no matter what products I use” or “easily get all particles, even the tiniest, out from between my teeth.”

Because the identification of customer needs requires a deep understanding of a customer’s experience, traditional methods rely on direct input from customers. Common methods to obtain direct input, such as experiential interviews and focus groups, are expensive and time-consuming, often resulting in delays in time to market. Although some firms use heuristics, such as managerial judgment or the review of web-based product comparisons, heuristic methods often miss new “excitement needs” or new strategic positionings.

User-generated content (UGC), with its extensive rich textual content, is a promising source from which to identify customer needs. UGC is available quickly and at low incremental cost to the firm. In many categories, UGC is extensive—for example, there are over 300,000 reviews on health and personal care products on Amazon alone. If UGC can be mined for abstract customer needs, UGC has the potential to identify as many, or perhaps more, customer needs than direct customer interviews. UGC provides additional advantages: (1) it is updated continuously enabling the firm to update its understanding of customer needs and (2) unlike customer interviews, firms can return to UGC at low cost to explore new insights further.

To use UGC, we must overcome three challenges. First, much UGC is repetitive and not relevant. Sentences such as “I highly recommend this product” do not express customer needs. We expect, and our analysis confirms, that most of the UGC concentrates on a relatively few customer needs. Second, the very scale of UGC makes it difficult for human readers to process. Third, UGC data are unstructured and mostly text-based. To identify abstract customer needs, we must be able to identify and understand rich structure.

Our goals in this paper are two-fold. First, we examine whether a reasonable corpus of UGC provides sufficient content to identify customer needs. We construct and analyze a custom dataset in which a professional marketing consulting firm provided a complete coding of (a) experiential interview transcripts and (b) a sample of sentences from Amazon reviews in the oral-care category. Second, we develop and evaluate a machine-learning hybrid approach to identify customer needs from UGC. We use machine learning to identify relevant content and remove redundancy from a large UGC corpus, and then rely on human judgment to formulate customer needs from selected content. We draw on recent research in deep learning, in particular, convolutional neural networks (CNN; Collobert, et al. 2011; Kim 2014) and dense word and sentence embeddings (Mikolov, et al. 2013a; Socher, et al. 2013). The CNN filters out non-informative content from a large UGC corpus. Dense word and sentence embeddings embed semantic content in a real-valued vector space. We use the properties of the sentence embeddings to sample a diverse set of non-redundant sentences for manual review. Manual review remains necessary in the last step because of the complex nature of abstract customer needs.

Our comparisons suggest that, if we limit human effort to that required to review experiential interviews, then UGC provides a comparable set of customer needs to those obtained from experiential interviews. Even with this human effort constraint, UGC provides customer needs not identified from experiential interviews, although a few customer needs are missed. When we relax the human-effort constraint for reviewing sentences, but maintain effort less than would be required to interview and review, then UGC is a better source of customer needs. We further demonstrate that machine learning helps to eliminate non-informative and redundant content and, hence, makes manual review more efficient.

2. Related Research

2.1. Traditional Methods to Identify Customer Needs (and Link Needs to Product Attributes)

Given a set of customer needs, product-development teams use a variety of methods, such as quality function deployment, to identify customer solutions or product attributes that address customer needs (Akao 2004; Hauser and Clausing 1988; Sullivan 1986). For example, Chan and Wu (2002) review 650 research articles that develop, refine, and apply QFD to map customer needs to solutions. Zahay, Griffin and Fredericks (2004) review the use of customer needs in the "fuzzy front end," product design, product testing, and product launch. Customer needs can also be used to identify attributes for conjoint analysis (Green and Srinivasan, 1978; Orme, 2006). Kim, et al. (2017) propose a benefit-based conjoint-analysis model which maps product attributes to latent customer needs before estimation.

Researchers in marketing and engineering have developed and refined many methods to elicit customer needs directly from customers. The most common methods rely on focus groups, experiential interviews, or ethnography as input. Trained analysts then review the input, manually identify customer needs, remove redundancy and structure the customer needs (Alam and Perry, 2002; Goffin, et al. 2012; Kaulio 1998). Some researchers augment interviews with structured methods such as repertory grids (Wu and Shich 2010). Recently, Schaffhausen and Kowlewski (2015; 2016) proposed using a web interface to ask customers to enter customer needs and stories directly. They then rely on human judgment to structure the needs and remove redundancy.

Typically, customer-need identification begins with 20-30 qualitative experiential interviews. Multiple analysts review transcripts, highlight customer needs and remove redundancy ("winnowing") to produce a basic set of approximately 100 abstract customer-need statements. Affinity groups or clustered customer-card sorts then produce a hierarchy of primary, secondary, and tertiary needs for product development and strategic positioning (Griffin and Hauser 1993; Jiao and Chen 2006). These methods are often called voice-of-the-customer (VOC) methods.

2.2. UGC Text Analysis in Marketing and Product Development

Researchers in marketing have developed a variety of methods to mine unstructured textual data to address managerial questions. See reviews in Büschken and Allenby (2016) and Fader and Winer (2012). The research closest to our goals uses word co-occurrences and variations of Latent Dirichlet Analysis to characterize customer preferences or latent topics in product discussions (Archak, Ghose, and Panagiotis 2016; Büschken and Allenby 2006; Lee and Bradlow 2011; Tirunillai and Tellis 2014; Netzer, et al. 2012). In engineering, product attribute elicitation literature is close to the goals of our paper, although the focus is on attributes rather than more-abstract customer needs. Jin, et al. (2015), Kuehl (2016), and Peng, Sun, and Revankar (2012) propose automated methods to map customer feedback into engineering characteristics. These papers focus on particular parts of speech or manually identified word combinations and use clustering techniques or Latent Dirichlet Analysis to identify product attributes and levels to be considered in product development. Our methods augment these literatures by focusing on the more-abstract deeper semantic structure of customer needs.

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. A CNN is a supervised prediction technique which is particularly suited to natural language processing tasks. A CNN often contains multiple layers which convolve numerical representations of sentences to create features for a logit model, which predicts the final outcome. CNNs demonstrate state-of-the-art performance with minimum tuning in such problems as relation extraction (Nguyen and Grishman 2015), named entity recognition (Chiu and Nichols 2015), and sentiment analysis (dos Santos and Gatti 2014).

Dense word and sentence embeddings are real-valued vector mappings (typically 20-300 dimensions), which are trained such that vectors for similar words (or sentences) are close in the vector space. The theory of dense embeddings is based on the Distributional Hypothesis, which states that words that appear in a similar context share semantic meaning (Harris, 1954). High-quality word and sentence embeddings can be used as an input for downstream NLP applications and models (Lample, et al. 2016; Kim 2014). Somewhat unexpectedly, high-quality word embeddings capture not only semantic similarity, but also semantic relationships (Mikolov, et al. 2013b). Using the convention of bold type for vector, then if $\boldsymbol{v}('\mathrm{word}')$ is the word embedding for 'word,' Mikolov et al. (2013b) demonstrate that word embeddings trained on the Google News Corpus have the following properties:

$$\boldsymbol{v}(\mathrm{king}) - \boldsymbol{v}(\mathrm{man}) + \boldsymbol{v}(\mathrm{woman}) \approx \boldsymbol{v}(\mathrm{queen})$$

$$\boldsymbol{v}(\mathrm{walking}) - \boldsymbol{v}(\mathrm{swimming}) + \boldsymbol{v}(\mathrm{swam}) \approx \boldsymbol{v}(\mathrm{walked})$$

$$\boldsymbol{v}(\mathrm{Paris}) - \boldsymbol{v}(\mathrm{France}) + \boldsymbol{v}(\mathrm{Italy}) \approx \boldsymbol{v}(\mathrm{Rome})$$

We train word embeddings using a large unlabeled corpus of online reviews. We then apply the trained word embeddings to enhance the performance of the CNN and to avoid repetitiveness among the sentences selected for manual review.

3. A Proposed Machine Learning Hybrid Method to Identify Customer Needs

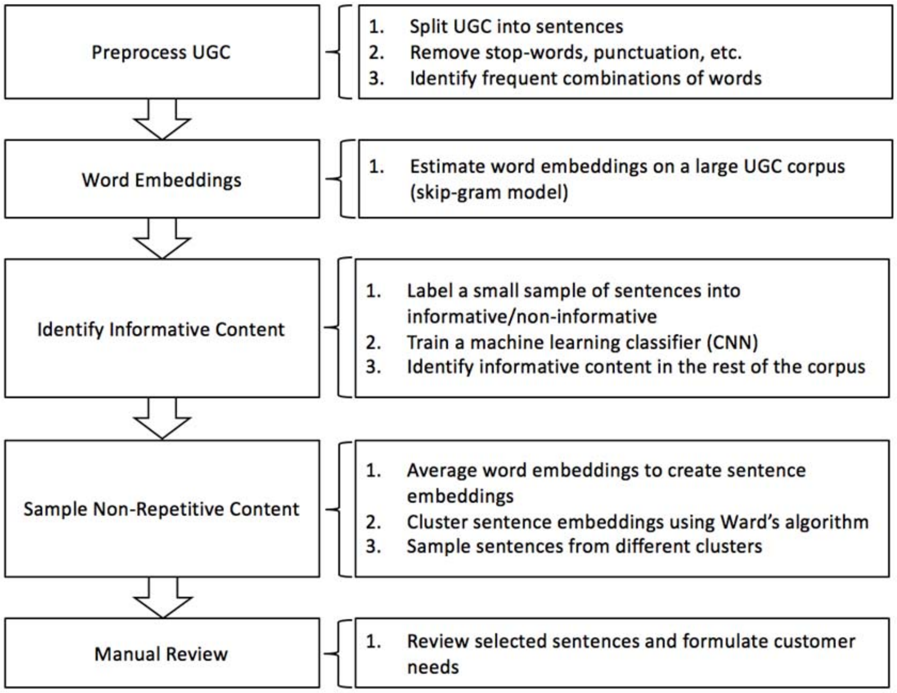
We propose a method that uses machine learning to screen UGC for sentences rich in a diverse set of customer needs. Identified sentences are then reviewed by human analysts to formulate precise customer needs. Machine-human hybrids have proven effective in a broad set of applications. For example, Qian, et al. (2012) combine machine learning and human judgment to locate research when authors' names are ambiguous (e.g., there are 117 authors with the name Lei Zhang). Supervised learning identifies clusters of similar publications and human readers associate authors with the clusters. The resulting hybrid is more accurate than machine learning alone and more efficient than human classification. Colson (2016) describes Stitch Fix's machine-human hybrid in which machine learning helps create a short list of apparel from vast catalogues, then human curators make the final recommendations to consumers.

Figure 1 summarizes our approach. The proposed method consists of five stages:

1. **UGC**. We harvest readily available UGC from either public sources or propriety company databases. We split UGC into sentences, eliminate stop-words, numbers and punctuation, and concatenate frequent combinations of words.
2. **Word Embeddings**. We train word embeddings using a skip-gram model on preprocessed UGC sentences, and use word embeddings as an input in the following stages.
3. **Identify Informative Content**. We train and apply a CNN classifier to filter out non-informative sentences so that the remaining corpus is rich in informative content.
4. **Sample Diverse Content**. We cluster sentence embeddings and sample sentences from different clusters to select a set of sentences likely to represent diverse customer needs.
5. **Final Extraction Of Representative Customer Needs**. Analysts review the diverse, informative sentences to identify customer needs.

The proposed architecture achieves the same goals as the traditional voice-of-the-customer approach. The preprocessed UGC replaces experiential interviews, the automated sampling of informative sentences is analogous to manual highlighting of sentences, and the clustering of word embeddings is analogous to manual winnowing to remove redundancy.

Figure 1 Systems Architecture for Identifying Customer Needs from UGC


Preprocess UGC
1. Split UGC into sentences
2. Remove stop-words, punctuation, etc.
3. Identify frequent combinations of words
Word Embeddings
1. Estimate word embeddings on a large UGC corpus (skip-gram model)
Identify Informative Content
1. Label a small sample of sentences into informative/non-informative
2. Train a machine learning classifier (CNN)
3. Identify informative content in the rest of the corpus
Sample Non-Repetitive Content
1. Average word embeddings to create sentence embeddings
2. Cluster sentence embeddings using Ward's algorithm
3. Sample sentences from different clusters
Manual Review
1. Review selected sentences and formulate customer needs

3.1. Stage 1: Preprocessing UGC to Identify Sentences within the UGC

Prior experience in the manual review of UGC by professional analysts, suggests that sentences are most likely to contain customer needs and are a natural unit by which analysts process experiential interviews and UGC. We preprocess raw UGC to transform the UGC corpus into a set of sentences. We use an unsupervised sentence tokenizer from the natural language toolkit (Kiss and Strunk 2006). We automatically eliminate stop-words (e.g., ‘the’ and ‘and’) and non-alphanumeric symbols (e.g., question marks and apostrophes), and transform numbers into number signs and letters to lower case.

We join words that appear frequently together with the ‘_’ character. For example, in oral care, the bigram ‘Oral B’ is treated as a combined word pair, ’oral_b.’ We join words ‘a’ and ‘b’ into a single phrase if they appear together relatively often in the corpus. The specific criterion is:

$$\frac{count(a,b)-\delta}{count(a)\cdot count(b)}\cdot M>\tau$$

where M is the total vocabulary size. The tuning parameter, δ, prevents concatenating very infrequent words, and the tuning parameter, τ, is balanced so that the number of bigrams is not too few or too many for the corpus. Both parameters are set by judgment. For our proof of concept, we set $(\delta, \tau) = (5,10)$.

We drop sentences that are too short (less than four words after preprocessing) or too long (more than fourteen words after preprocessing). The bounds are selected to drop approximately 10% of the shortest and 10% of the longest sentences.

3.2. Stage 2: Training Word Embeddings with a Skip-Gram Model

Word embeddings incorporate contextual information about words and serve as an input to Stages 3 and 4 (Baroni, Dinu, and Kruszewski, 2014). We train word embeddings on the preprocessed UGC corpus using a skip-gram model (Mikolov, et al. 2013a). The skip-gram model is a predictive model which maximizes the average log-likelihood of words appearing together in a sequence of c words. Specifically, if I is the number of words in the corpus, V is the set of all feasible words in the vocabulary, and $\boldsymbol{v}_i$ are 20-dimensional real-vector word embeddings, we select the $\boldsymbol{v}_i$ to maximize:

$$\frac{1}{I}\sum_{i=1}^{I} \sum_{\substack{-c \le j \le c \\ j \ne 0}} log\, p\left(word_{i+j} \middle| word_i\right)$$

$$p\left(word_j \middle| word_i\right) = \frac{exp\left(\boldsymbol{v}_j \boldsymbol{v}'_i\right)}{\sum_{k=1}^{|V|} exp\left(\boldsymbol{v}_k \boldsymbol{v}'_i\right)}$$

To make calculations feasible, we use ten-word negative sampling to approximate the denominator in the conditional probability function (Mikolov et al., 2013b). For our proof of concept, we use $c = 5$.

The trained word embeddings in our proof-of-concept study capture semantic meaning in oral care. For example, ‘recommend’, ‘would_recommend’, ‘highly_recommend’, ‘reccommend’, and ‘recommed’ are close in the vector space. Similarly, the three words closest to ‘toothbrush’ are ‘pulsonic’, ‘sonicare’ and ‘tb’, with the last being a commonly-used abbreviation for toothbrush.

3.3. Stage 3: Identifying Informative Sentences with a Convolutional Neural Network (CNN)

Depending on the corpus, UGC can contain substantial amounts of content that does not represent customer needs. Such content includes evaluations, complaints, and non-informative lists of features such as "This product can be found at CVS." Machine learning improves the efficiency of manual review by eliminating non-informative content. For example, suppose that only 40% of the sentences are informative in the corpus, but after machine learning screening, 80% are informative. Analysts can identify customer needs much more efficiently by focusing on a sample rich in informative sentences. To train the machine learning classifier, some sentences must be labeled manually. However, there are efficiency gains because such labeling is much easier than formulating customer needs from informative sentences.

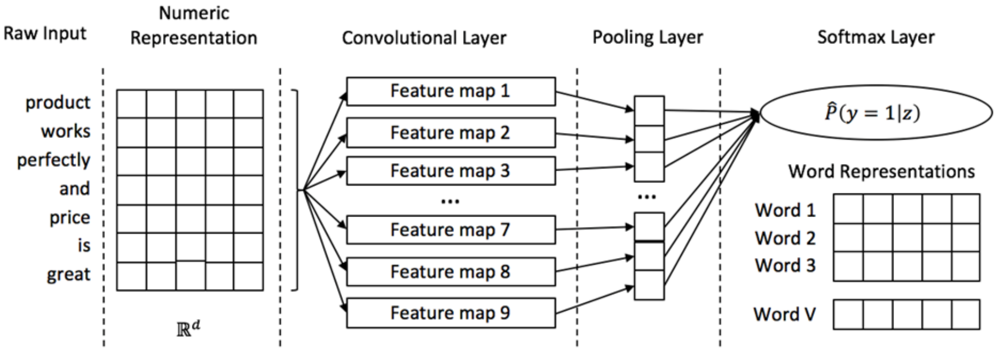
We use a convolutional neural network (CNN) to identify informative sentences. A major advantage of the CNN is that CNNs quantify raw input automatically and endogenously based on the training data. CNNs apply a combination of convolutional and pooling layers to word representations to generate "features," which are then used to make a prediction. The "features" generated by CNNs are similar to dependent variables in OLS. ("Features" in the CNN should not be confused with product features.) In contrast, a support-vector machine (SVM) depends critically on the handcrafted features, which are the transformations of the raw data designed by researchers to improve prediction in a particular application. High-quality SVM features require substantial human effort. CNNs have been proven to provide comparable performance to handcrafted SVMs without substantial application-specific human effort (Kim 2014; Lei, et al. 2015).

A typical CNN consists of multiple layers. Each layer has hyperparameters, such as the number of filters and the size of the filters. We custom select these hyperparameters, and the number and type of layers, by cross-validation. Each layer also has numerical parameters, such as the parameters of the filters used in the convolutional layers. These parameters are calibrated during training. We train the

CNN by selecting the parameter values that maximize the CNN's ability to label sentences as informative vs. non-informative.

Figure 2 illustrates the architecture of the CNN in our application. We stack a convolutional layer, a pooling layer, and a softmax layer. The convolutional layer generates nine feature maps. This specification modifies Kim's (2014) architecture for sentence classification task to account for less training data.

Figure 2 Convolutional Neural Network Architecture



3.3.1. Numerical Representations of Words for Use in the CNN

For every word in the text corpus, the CNN stores a numerical representation of the word. Numerical representations of words are the real vector parameters of the model which are calibrated to improve prediction. To facilitate training of the CNN, we initialize representations with word embeddings from Stage 2. However, we then allow the CNN to update the numerical representations to enhance predictive ability (Lample, et al. 2016).

The CNN quantifies sentences by concatenating word embeddings. If $\boldsymbol{v}_i$ is the word embedding for the i^{th} word in the sentence, then the sentence is represented by a vector $\boldsymbol{v}$

$$\boldsymbol{v} = [\boldsymbol{v}_1, \dots, \boldsymbol{v}_n] \in \mathbb{R}^{d \times n}$$

where n is the number of words in the sentence and $d = 20$ is the dimensionality of the word embeddings.

3.3.2. Convolutional Layer

Convolutional layers create multiple feature maps by applying convolutional operations with varying filters to the sentence representation. A filter is a real-valued vector, $\boldsymbol{w}_t \in \mathbb{R}^{d \times h_t}$, where h_t is a size of the filter. Filters are applied to different parts of the vector $\boldsymbol{v}$ to create feature maps:

$$\boldsymbol{c}^t = [c_1^t, \dots, c_{n-h_t+1}^t]$$

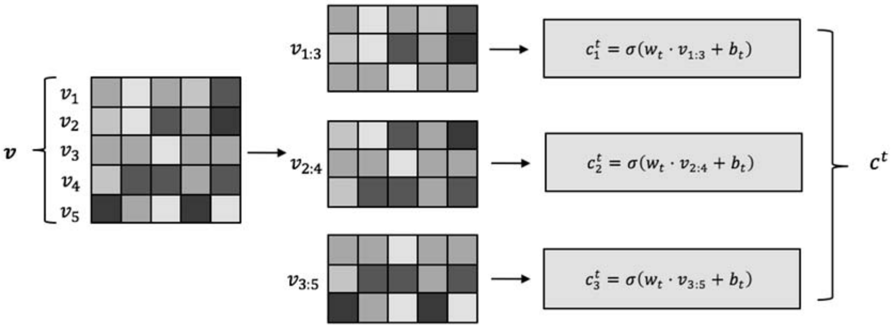
$$c_i^t = \sigma\big(\boldsymbol{w}_t \cdot \boldsymbol{v}_{i:i+h_t-1} + b_t\big)$$

where t indexes the feature maps, $\sigma(\cdot)$ is a non-linear activation function, $\sigma(x) = \max(0, x)$, $b_t \in \mathbb{R}$ is a bias, and $\boldsymbol{v}_{i:i+h_t-1}$ is a concatenation of representations of words i to $i + h_t - 1$ in the sentence:

$$\boldsymbol{v}_{i:i+h_t-1} = [\boldsymbol{v}_i, \dots, \boldsymbol{v}_{i+h_t-1}]$$

We consider filters of the size $h_t \in \{3, 4, 5\}$, and use three filters of each size. The number of filters and their size are selected with cross-validation. The numerical values for filters, $\boldsymbol{w}_t$, and biases, b_t, are calibrated when the CNN is trained. As an illustration, Figure 3 show how a feature map is generated with a filter of size, $h_t = 3$.

Figure 3. Example Feature Map, $\boldsymbol{c}^t$ Generated with a Filter, $\boldsymbol{w}_t$, of Size $\boldsymbol{h}_t = 3$.


v
v_1
v_2
v_3
v_4
v_5
$v_{1:3}$
$v_{2:4}$
$v_{3:5}$
$c_1^t = \sigma(w_t \cdot v_{1:3} + b_t)$
$c_2^t = \sigma(w_t \cdot v_{2:4} + b_t)$
$c_3^t = \sigma(w_t \cdot v_{3:5} + b_t)$
c^t

3.3.3. Pooling Layer

The pooling layer transforms feature maps into shorter vectors. The role of the pooling layer is to reduce dimensionality of the output of the convolutional layer to be used in the next layer. Pooling to the k^{th} largest features or simply using the largest feature has proven effective in NLP applications (Collobert, et al. 2011). We selected $k = 1$ with cross-validation. The output of the pooling layer is a vector, $\boldsymbol{z}$**,** that summarizes the results of pooling operators applied to the nine feature maps:

$$z_t = max[c_1^t, \dots, c_{n-h_t+1}^t]$$

$$\boldsymbol{z} = [z_1, z_2, \dots, z_9]$$

The vector, $\boldsymbol{z} \in \mathbb{R}^9$, is now an efficient numerical representation of the sentence and can be used to classify the sentence as either informative or not informative.

3.3.4. Softmax Layer

The final layer of the CNN is called the softmax layer. The softmax layer transforms the output of the pooling layers, $\boldsymbol{z}$, into a probabilistic prediction of whether the sentence is informative ($y = 1$) or not informative ($y = 0$). Marketing researchers will recognize the softmax layer as a binary logit model which uses the $\boldsymbol{z}$ vector as explanatory variables. The estimate of the probability that the sentence is informative, $P(y = 1|\boldsymbol{z})$, is given by:

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

The parameters of the logit model, $\boldsymbol{\theta}$, are determined when the CNN is trained. In our application, we declare a sentence to be informative if $P(y = 1|\boldsymbol{z}) > 0.5$, although other criteria could be used and tuned to a target tradeoff between false positives and false negatives.

3.3.5. Calibration of the Parameters of the CNN

For our application we calibrate the nine filters, $\boldsymbol{w}_t \in \mathbb{R}^{d \times h_t}$, and the nine biases, b_t, in the convolutional layer, and the vector $\boldsymbol{\theta}$ in the softmax layer. In addition, we fine tune the word embeddings, $\boldsymbol{v_i}$, to enhance the ability of the CNN's predictions (e.g., Kim 2014). We calibrate all parameters simultaneously by minimizing the cross-entropy error on the training set of manually labeled sentences ($\boldsymbol{w}$ is a concatenation of the $\boldsymbol{w}_t$'s):

$$\widehat{\boldsymbol{w}}, \widehat{\boldsymbol{b}}, \widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{v}} = argmax_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{v}} L(\boldsymbol{w}, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{v})$$

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

N is a size of training set, y_n are the manually assigned labels, and $\hat{y}_n$ is the predictions of the CNN. We solved the optimization problem iteratively with the RMSProp optimizer on mini-batches of size 32 and a drop rate of 0.3. Optimization terminated when the cross-entropy error on the validation set did not decrease over 5 consecutive iterations. See Tieleman and Hinton (2012).

3.3.6. Evaluating the Performance of the CNN

We evaluate the quality of a classifier using two common criteria: precision and recall (Lee and Bradlow, 2011):

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

where TP, FP and FN correspond to a number of identified true positives (*TP*), false positives (*FP*), and false negatives (*FN*) in a holdout set. Precision is interpreted as the share of informative sentences among the sentences identified as informative and recall is a share of informative sentences correctly identified. 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).

$$F_1 = \frac{precision \cdot recall}{\frac{1}{2}(precision + recall)}$$

3.4. Stage 4: Clustering Sentence Embeddings and Sampling to Reduce Redundancy

UGC is repetitive and often focuses on a small set of customer needs. Manual review of repetitive content is inefficient. Moreover, repetitiveness makes the manual review onerous and boring for analysts, causing analysts to miss excitement customer needs that are mentioned rarely. If the analysts miss excitement customer needs, then the firm misses valuable new product opportunities and/or strategic positionings. To avoid repetitiveness, we seek to "span the set" of customer needs. We construct sentence embeddings which encode semantic relationships between sentences, and use sentence embeddings to reduce redundancy by sampling content for manual review from maximally different parts of the space of sentences. (Sentence embeddings are related to, but not the same as, the word concatenations used in the CNN.)

Researchers often create sentence embeddings by taking a simple average of word embeddings corresponding to the words in the sentence (Iyyer et al., 2015), explicitly modeling semantic and syntactic structure of the sentences with neural methods (Tai, Socher and Manning 2015), or training sentence embeddings together with word embeddings (Le and Mikolov, 2014). Because averaging demonstrates similar performance to other methods and is both scalable and transferable (Iyyer et al., 2015), we use averaging in our proof-of-concept application.

Being the average of word embeddings, sentence embeddings represent semantic similarity among sentences. Using this property, we group sentences into clusters. We choose Ward's hierarchical clustering method because it is commonly used in VOC studies (Griffin and Hauser 1993), and other areas of marketing research (Dolnicar 2003). We set the number of clusters equal to the number of sentences which analysts choose to review, and sample one sentence randomly from each cluster.

3.5. Stage 5: Manually extract customer needs

To achieve high precision in formulating abstract customer needs, the final extraction of customer needs is best done by trained analysts. We evaluate in §5 whether manual extraction becomes more efficient using informative, diverse sentences identified with the CNN and sentence-embedding clusters.

4. Oral Care Proof-of-Concept: Evaluation of UGC's Potential

We use empirical data to examine two questions. (§4) Does UGC contain sufficient raw material from which to abstract customer needs? And (§5) Do each of the machine-learning steps enhance efficiency? We address both questions with a custom dataset in the oral-care category. We selected oral care because oral-care customer needs are sufficiently varied, but not so numerous as to overcomplicate comparisons. As a proof-of-concept test, our analyses establish a key example. We discuss applicability to other categories in §6.

4.1. Baseline Comparison: Experiential Interviews in Oral Care

We obtained a detailed set of customer needs from an oral care voice-of-the-customer (VOC) analysis that was undertaken by a professional market research consulting firm. The firm has almost thirty years of VOC experience spanning hundreds of successful product-development applications across a wide-variety of industries. The oral-care VOC provided valuable insight to the client and led to successful new products. The VOC was based on standard methods: experiential interviews, with sentences highlighted by experienced analysts aided by the firm's proprietary software. After winnowing, customer needs were clustered by a customer-based affinity group. The output is six primary customer needs, 22 secondary customer needs, and 86 tertiary customer needs. Table 1 summarizes the primary and secondary needs. Although the primary needs are short phrases, many secondary needs are more elaborate and abstract: "tools are easy to maneuver and manipulate." Tertiary needs (not shown in Table 1) are even more elaborate. Examples include: "Oral care products that do not create any odd sensations in my mouth while using them (e.g. tingling, burning, etc.)" or "My teeth feel smooth when I glide my tongue over them." Such customer needs are more than their component words; they describe a desired outcome in the language that the customer uses to describe the desired outcome.

Table 1. Voice of the Customer for Oral Care as Obtained from Experiential Interviews. Only primary and secondary needs are shown. There were 86 tertiary needs.

Primary Customer Needs	Secondary Customer Needs
Feel Clean And Fresh (Sensory)	Clean Feeling in My Mouth Fresh Breath All Day Long Pleasant Taste and Texture
Strong Teeth And Gums	Prevent Gingivitis Able to Protect My Teeth Whiter Teeth
Product Efficacy	Effectively Clean Hard to Reach Areas Gentle Oral Care Products Oral Care Products that Last Tools are Easy to Maneuver and Manipulate
Knowledge And Confidence	Knowledge of Proper Techniques Long Term Oral Care Health Motivation for Good Check-Ups Able to Differentiate Products
Convenience	Efficient Oral Care Routine (Effective, Hassle-Free and Quick) Oral Care “Away From the Bathroom”
Shopping / Product Choice	Faith in the Products Provides a Good Deal Effective Storage Environmentally Friendly Products Easy to Shop for Oral Care Items Product Aesthetics

4.2. Fully-Coded UGC Data from the Oral-Care Category

To compare UGC to experiential interviews, we needed a complete coding of a sample of a UGC corpus. From the 115,099 oral-care reviews on Amazon spanning the period from 1996 to 2014, we randomly sampled 12,000 sentences split into an initial set of 8,000 sentences and a second set of 4,000 sentences (McAuley, et. al. 2015). To maintain a common level of training and experience for reviewing UGC and experiential interview transcripts, we recruited three experienced analysts from the same firm that provided the interview-based VOC.

We chose 8,000 sentences for our primary evaluation because the analysts judged that the effort to review 8,000 sentences would be comparable, albeit slightly less than, the effort to review a typical set of experiential-interview transcripts. For the first 8,000 sentences, the analysts fully coded every sentence to determine whether it contained a customer need and, if so, whether the customer need could be mapped to a customer need identified by the VOC, or whether the customer need was a newly identified customer need. This unique custom data set enables us to compare the customer needs obtained from UGC using analyst effort that is comparable to the effort allocated to reviewing interview transcripts. For the second 4,000 sentences, the analysts coded the UGC for those customer needs which were identified in the experiential interviews but not in the first 8,000 sentences.

The goal of the second set of sentences was to determine whether there were any customer needs that could not be identified from UGC. As a constraint, we wanted the analyst effort for both sets of sentences to be substantially less than they effort required for the interview-based VOC. The professional market research firm reports that the effort, required to review, highlight, and winnow customer needs, is about 40%-55% of the effort required to schedule and interview customers. Thus, analysts could code 13,000 to 17,000 UGC sentences for the amount of human effort allocated to an interview-based VOC. The analysts confirmed that the effort to code both the first and second set of sentences would still be less than the effort necessary for a traditional VOC. (Full coding consumes substantial time from highly-paid and sought-after analysts. At industry rates, we would require $50,000 to $75,000 in professional services to obtain our custom datasets.)

4.3. Descriptive Statistics and Comparisons

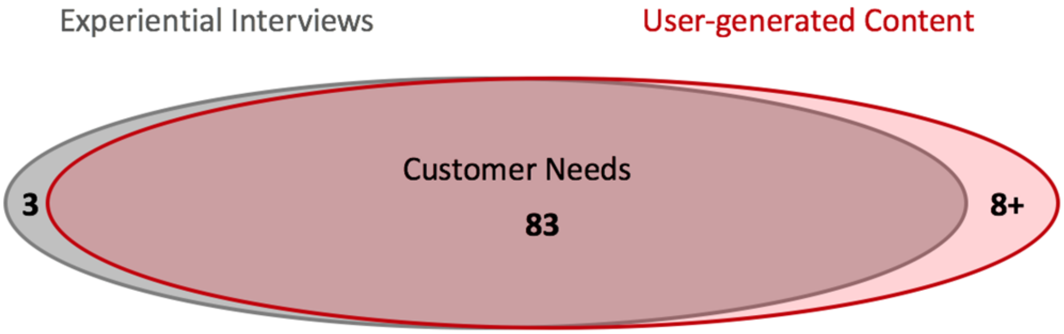
Oral-care UGC was particularly rich in content; 52% of the 8,000 sentences contained at least one customer need and 9.2% of the sentences contained two or more customer needs. However, the corpus was highly repetitive; 10% of the most frequent customer needs were articulated in 54% of the informative sentences. On the other hand, 17 customer needs were articulated no more than 5 times in the corpus of 8,000 sentences.

We consider first the case where analysts allocate at most as much time coding UGC as they would have allocated to code experiential interviews. Because we are interested in the potential of the

UGC corpus, we do not yet exploit machine-learning efficiencies. In this scenario, analysts identified 74 of the 86 tertiary experiential-interview-based customer needs, but also identified an additional 8 needs.

The second scenario expands the human effort allocated to coding but still maintains the effort to be below the human effort to interview respondents and code the transcripts. In the second set of 4,000 sentences, the analysts were able identify 9 of 12 missing customer needs. With 12,000 sentences, that brings the total to 83 of the 86 experiential-interview-based customer needs and 91 of the 94 total needs (97%). We did not code for customer needs other than the 12 missing needs. Had we had the resources to do so, we would likely have increased the number of UGC-based incremental customer needs. Overall, analysts identified (at least) 91 customer needs from UGC and 86 from experiential interviews. These results are summarized in Figure 4. In summary, at least in oral care, analyzing UGC has the potential to identify more customer needs at a lower overall cost.

Figure 4. Comparison of Customer Needs from Experiential Interviews and UGC



4.4. Prioritization of Customer Needs

To address whether the eight incremental UGC customer needs and/or the three incremental experiential-interview customer needs were important, we conducted a prioritization survey. We randomly selected 197 customers from a professional panel (PureSpectrum), screened for interest in oral care, and asked customers to rate the importance of each tertiary customer need on a 0 to 100 point scale. Customers also rated whether they felt that their current oral-care products performed well on these customer needs. Such measures are used commonly in VOC studies and have proven to provide valuable insight for product development (Corrigan 2013).

Using Corrigan's (2013) classification, the UGC-unique customer needs identify 20% of the hidden opportunities and the interview-unique needs identify 8% of the hidden opportunities. (Corrigan assumes the customers sometimes evaluate needs below the medians on importance and performance when customers anticipate that no current product fulfills those customer needs.) For example, two UGC-unique hidden opportunities are "An oral-care product that does not affect my sense of taste," and "An oral care product that is quiet." An interview-based hidden opportunity is "Oral care tools that can easily used by left-handed people."

In summary, UGC identifies the vast majority of customer needs (97%) and the vast majority of hidden opportunities (92%). UGC-unique needs identify at least seven hidden opportunities while interview-only needs identify two hidden opportunities.

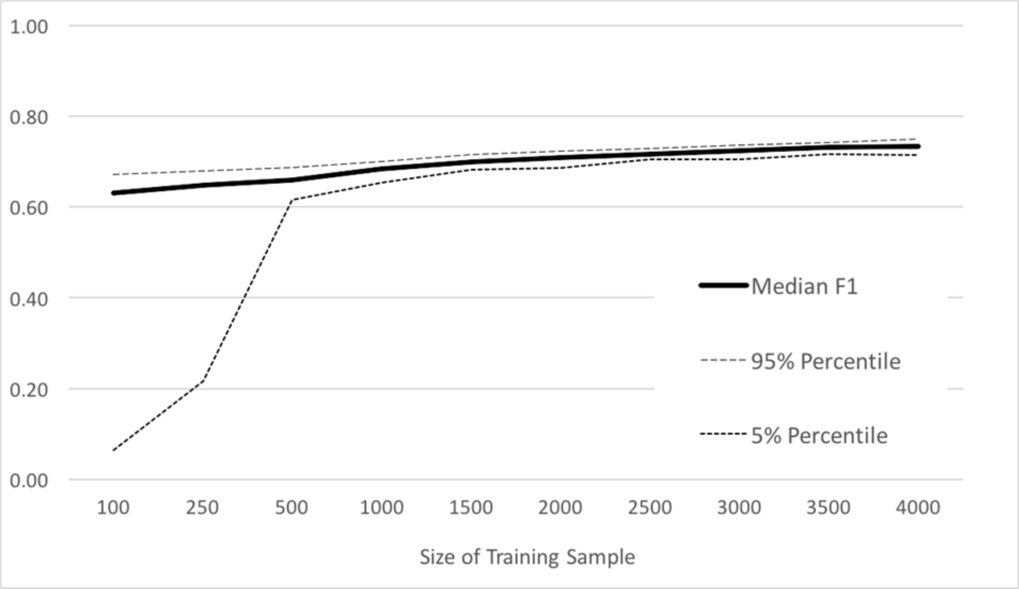
5. Oral Care Proof-of-Concept: Evaluation of Machine-Human Hybrid Method

5.1. CNN to Eliminate Non-Informative Sentences

There is a tradeoff to be made when training a CNN. With a larger training sample, the CNN is better at identifying informative content, but there is an opportunity cost to using analysts to classify informative sentences. Fortunately, labeling sentences is faster and easier than abstracting needs from sentences. Nonetheless, we must consider analyst effort when we select the size of the training sample.

Figure 5 plots the F_1-score of the CNN as a function of the size of the training sample. We conduct 100 iterations where we randomly draw a training set, train the CNN with the architecture described in §3.3, and measure performance on the test set. Figure 5 suggests that performance of the CNN stabilizes after 500 training sentences, with some slight improvement after 500 training sentences.

Figure 5. F_1 **score as a Function of the Size of the Training Sample**



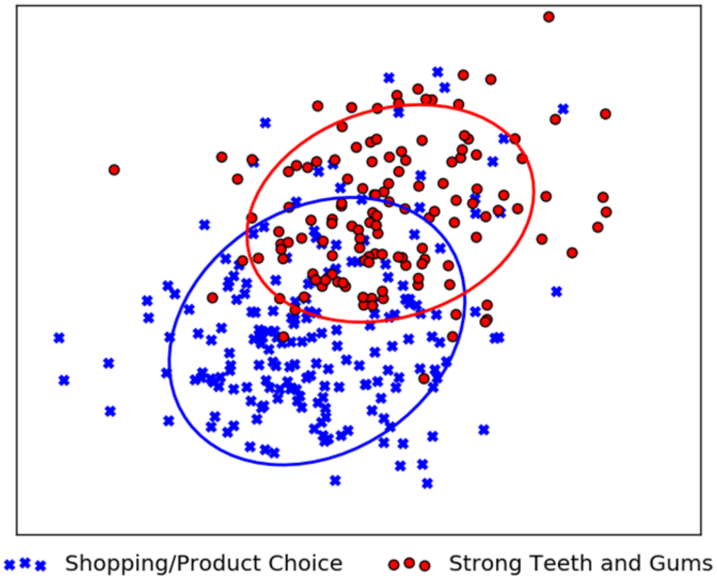
To be effective, the CNN should identify informative sentences that contain customer needs that were not in the sentences in the training set. In our case, the CNN identifies sentences that contain customer needs that were "new," that is, not in the sentences used to train the CNN. For example, the CNN identifies an informative sentence that contains the "new" customer need, "using as few products as possible and still having an effective oral care routine." In the prioritization survey, this customer need was in the upper quartile of importance.

5.2. Clustering Sentence Embeddings to Reduce Redundancy

To visualize whether or not sentence-embedding clusters reduce redundancy, we use principle components analysis to project the sentence embeddings onto two dimensions. Information is lost when we project from 20 dimensions to two dimensions, but the two-dimensional plot enables us to visualize whether or not customer needs were separated by the clusters. As an illustration, Figure 6 reports the two-dimensional projection for two primary needs. Figure 6 is typical; we get similar results for other customer needs.

The red dots are the projections of sentence embeddings that were coded (by analysts) as belonging to the primary customer need: "strong teeth and gums." The blue crosses are sentence embeddings that were coded as "shopping/product choice." (Review Table 1.) The ovals represent the smallest ellipsis inscribing 90% of the corresponding set. Figure 6 suggests that, while not perfect, the clusters of sentence embeddings achieved separation among customer needs and, hence, are likely to reduce redundancy and enable analysts to identify a diverse set of customer needs.

Figure 6. Two-dimensional Projection of 20-Dimensional Sentence Embeddings



5.3. Gains in Efficiency Due to Machine Learning

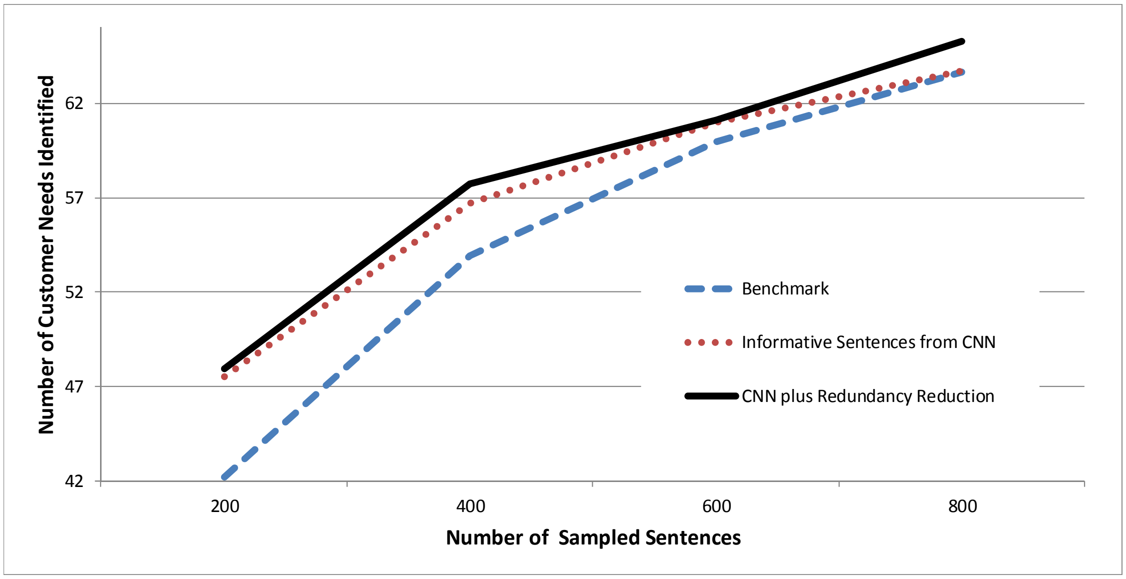
We seek to determine whether the machine-learning methods improve efficiency relative to current practice. Because we have a database of 8,000 completely coded sentences (6,700 after preprocessing), we can compute counterfactual simulations of the number of customer needs that would have been identified by various methods for various numbers of sampled UGC sentences. For these counterfactual simulations we want the CNN to be well-trained so that we evaluate the full potential of the CNN rather than a CNN that is limited by the size of the training sample. We choose a training sample of 3,700 sentences. At this level of training, the CNN achieves a precision of 74.4%, a recall of 72.8%, and an F_1 of 73.6%. The large training sample reduces randomness in our analysis, but the relative comparisons of methods are unlikely to change with smaller training samples. After training, there are 3,000 of the 6,700 sentences available for counterfactual simulations.

We compute counterfactuals for three methods assuming the firm has sufficient analyst resources to review Y sentences for $Y = \{200, 400, 600, 800\}$. The baseline method is current practice—a random draw from the corpus. The second method uses the CNN to identify informative sentences for review, but does not use sentence embeddings to reduce redundancy. The third method uses the sentence-clusters to reduce redundancy among sentences identified as informative by the CNN. For each method, and for each value of Y, we draw sufficient sentences from the corpus to provide Y sentences for review. We count the unique needs identified in the Y sentences and repeat the process 10,000 times to produce the plots in Figure 7.

The blue dashed line in Figure 7 reports benchmark performance. The CNN improves efficiency as indicated in the red dotted line. Using the CNN and clustering sentence embeddings increases efficiency further as indicated by the solid black line. Over the range of the counterfactual simulations, there are gains due to using the CNN to eliminate non-informative sentences and additional gains due to using sentence embeddings to reduce redundancy within the corpus.

We also interpret Figure 7 horizontally. The benchmark requires, on average, 832 sentences to identify 64.2 customer needs. If we prescreen with machine learning to select non-redundant informative sentences, an analyst can identify the same number of customer needs from approximately 700 sentences—84% of the sentences required by the baseline. The efficiencies are even greater at 200 sentences (71%) and 400 sentences (78%). At professional billing rates across many categories, this represents substantial time and cost savings. VOC customer-need identification methods has been optimized over almost thirty years of continuous improvement; we expect the machine-learning methods, themselves, to be subject to such continuous improvement as they are applied in the field.

Figure 7. Efficiencies among Various Methods to Select UGC Sentences for Review



6. Discussion, Summary, and Future Research

We addressed two questions: (1) Can UGC be used to identify abstract customer needs? And (2) can machine learning enhance the process? The answer to both questions is yes. Holding review-effort constant, UGC is a comparable source of customer needs to experiential interviews. Adding more review effort while saving substantial interviewing effort, UGC provides a richer set of customer needs with overall savings in time and effort. Machine learning makes the process more efficient. A CNN successfully eliminates non-informative content and clustering sentence embeddings reduces redundancy. Even in our proof of concept, machine learning gains are 16-29%.

Answering these questions is significant. Every year thousands of firms rely on voice-of-the-customer analyses to identify new opportunities for product development, to develop strategic positioning strategies, and to select attributes for conjoint analysis. Typically, VOC studies, while valuable, are expensive and time-consuming. Time savings, such as those made possible with machine learning applied to UGC, are extremely important to product development. In addition, UGC seems to contain customer needs not identified in experiential interviews. New customer needs mean new opportunities for product development and/or new strategic positioning.

While we are enthusiastic about UGC, we recognize UGC is not a panacea. UGC is readily available for oral care, but UGC might not be available for every product category. For example, consider specialized medical devices or specialized equipment for oil exploration. The number of customers for such products is small and such customers may not post reviews, blog, or tweets. On the other hand, UGC is extensive for complex products such as automobiles or cellular phones. Machine-learning efficiencies in such categories may be necessary to make the review of UGC feasible.

Further research will enhance our ability to identify abstract customer needs with UGC. For example,

- Deep neural networks and sentence embedding are active areas of research in the NLP community. We expect the performance of the proposed approach to improve significantly with new developments in machine learning.
- UGC is updated continuously. Firms might develop procedures to monitor UGC continuously. Sentence embeddings can be particularly valuable. For example, firms might concentrate on customer needs that are distant from established needs in the 20-dimenional vector space.
- Future developments might automate the final step, or at least enhance the ability of analysts to abstract customer needs from informative, non-redundant content.
- Other forms of UGC, such as blogs and Twitter feeds may be examined for customer needs.
- Field experiments might assess whether, and to what degree, abstract customer needs provide more insights for product development compared to lists of words.

References

Akao Y (2004) *Quality Function Deployment (QFD): Integrating customer requirements into product design*, (New York, NY: Productivity Press).

Archak N, Ghose A, Ipeirotis PG (2016) Deriving the pricing power of product features by mining consumer reviews, *Management Science*. 57(8): 1485-1509.

Alam I., Perry C. (2002) A customer-oriented new service development process. *Journal of services Marketing*. 16(6):515-534.

Baroni M, Dinu G, Kruszewski G (2014) Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, MD. 238-247.

Brown SL, Eisenhardt KM (1995) Product development: Past research, present findings, and future directions. *The Academy of Management Review*. 20(2):343-378.

Büschken, J, Allenby GM (2016) Sentence-based text analysis for consumer reviews. *Marketing Science*. 35(6):953-975.

Chan L-K, Wu M-L (2002) Quality Function Deployment: A literature Review. *European Journal of Operational Research*. 143:463-497.

Chiu JP, Nichols E (2016). Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics* 4:357–370.

Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Pavel K (2011) Natural language processing (almost) from scratch. *Journal of Machine Learning Research*. 12:2493-2537.

Colson E (2016) Human machine algorithms: Interview with Eric Colson. http://blog. fastforwardlabs.com/2016/05/25/human-machine-algorithms-interview-with-eric.html.

Corrigan KD (2013 Wise choice: The six most common product development pitfalls and how to avoid them. *Marketing News*. (September) 39-44.

Dolnicar S (2003) Using cluster analysis for market segmentation – typical misconceptions, established methodological weaknesses and some recommendation for improvement. *Australasian Journal of Market Research*. 11(2):5-12.

dos Santos CN, Gatti M (2014) Deep convolutional neural networks for sentiment analysis of short texts. *Proceedings the 25th International Conference on Computational Linguistics: Technical Papers.* Dublin, Ireland, 69–78,

Fader PS, Winer RS (2012) Introduction to the special issues on the emergence can impact of user-generated content. *Marketing Science*. 31(3):369-371.

Goffin K, Varnes CJ, van der Hoven C, Koners U (2012) Beyond the voice of the customer: Ethnographic market research. *Research Technology Management*. 55(4):45-53.

Green PE, Srinivasan V (1978) Conjoint analysis in consumer research: issues and outlook. *Journal of Consumer Research* 5(2):103-123.

Griffin A., Hauser JR (1993) The voice of the customer. *Marketing Science.* 12(1):1-27.

Griffin A, Price RL, Maloney MM, Vojak BA, Sim EW (2009) Voices from the field: how exceptional electronic industrial innovators innovate. *Journal of Product Innovation Management*. 26:222-240.

Hauser JR, Clausing D (1988) The house of quality. *Harvard Business Review*. 66(3):63-73.

Herrmann A, Huber F, Braunstein C (2000) Market-driven product and service design: Bridging the gap between customer needs, quality management, and customer satisfaction. *International Journal of Production Economics*. 66(1):77-96.

Iyyer M, Manjunatha V, Boyd-Graber J, Daumé III H. (2015) Deep unordered composition rivals syntactic methods for text classification. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing,* Beijing, China. 1:1681-1691.

Jiao J, Chen CH (2006) Customer requirement management in product development: a review of research issues. *Concurrent Engineering: Research and Applications*. 14(3):173-185.

Jin J, Hi P, Liu Y, and Lim SCJ (2015) Translating online customer opinions into engineering characteristics in QFD: A probabilistic language analysis approach. *Engineering Applications of Artificial Intelligence.* 41:115-127.

Kano N, Seraku N, Takahashi F, Tsuji S (1984) Attractive quality and must-be quality. *The Japanese Society for Quality Control* 14(2):39-48.

Kao Group (2016). http://www.company-histories.com/Kao-Corporation-Company-History.html.

Kaulio MA (1998) Customer, consumer and user involvement in product development: A framework and a review of selected methods. *Total Quality Management.* 9(1):141-149.

Kim Y (2014) Convolutional neural networks for sentence classification. *Empirical Methods in Natural Language Processing.* Doha, Qatar. 1746-1751.

Kim DS, Bailey RA, Hardt N, Allenby A (2017) Benefit-based conjoint analysis. *Marketing Science*, 36(1):54-69.

Kiss T, Strunk J (2006) Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485-525.

Krishnan V, Ulrich KT (2001) Product development decisions: A review of the literature. *Management Science.* 47(1):1-21.

Kuehl N (2016) Needmining: Towards analytical support for service design. *International Exploring Services Science*. 247:187-200.

Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition. *Proceedings of 2016 North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, CA:260-270.

Le QV, Mikolov T (2014) Distributed representations of sentences and documents. *Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, 32, 1188-1196.

Lee TY, Bradlow ET (2011) Automated marketing research using online customer reviews. *Journal of Marketing Research.* 48(5), 881-894.

Lei T, Barzilay R, Jaakkola T (2015) Molding CNNs for text: non-linear, non-consecutive convolutions. *Proceedings of 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal. 1565–1575.

Matzler K, Hinterhuber HH (1998) How to make product development projects more successful by integrating Kano's model of customer satisfaction into quality function deployment. *Technovation*. 18(1):25-38.

McAuley J, Pandey R, Leskovec J (2015) Inferring networks of substitutable and complementary products. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 785-794.

Mikolov T, Chen K, Corrado G, Dean J (2013a) Efficient estimation of word representations in vector space. arXiv:1301.3781v3 [cs.CL]m Sept 7,1301.3781.

Mikolov T, Sutskever I, Chen K., Corrado GS, Dean J (2013b) Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*. 26, 3111–3119.

Mikulić J, Prebežac D (2011). A critical review of techniques for classifying quality attributes in the Kano model. *Managing Service Quality.* 21(1):46-66.

Netzer O, Feldman R, Goldenberg J, Fresko M. (2012) Mine your own business: Market-structure surveillance through text mining. *Marketing Science.* 31(3), 521-543.

Nguyen TH, Grishman R (2015) Relation extraction: Perspective from convolutional neural networks. *Proceedings of North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies. Denver, CO. 39-48.

Orme BK (2006) *Getting started with conjoint analysis: Strategies for product design and pricing research, 2E.* (Madison WI: Research Publishers LLC).

Park CW, Jaworski BJ, MacInnis DJ (1986) Strategic brand concept-image management. *Journal of Marketing.* 50:135-145.

Peng W, Sun T, Revankar S (2012). Mining the `voice of the customer' for business prioritization. *ACM Transactions on Intelligent Systems and Technology.* 3 (2), 38:1-38-17.

Qian Y-N, Hu Y, Cui J, Nie Z (2001) Combining machine learning and human judgment in author disambiguation. *Proceedings of the 20th ACM Conference on Information and Knowledge Management*. Glasgow, United Kingdom.

Schaffhausen CR, Kowalewski TM (2015). Large-scale needfinding methods of increasing user-generated needs from large populations. *Journal of Mechanical Design.* 137(7): 071403.

Schaffhausen CR, Kowalewski TM (2016) Assessing quality of unmet user needs: effects of need statement characteristics. *Design Studies.* 44:1-27.

Socher R, Perelygin A, Wu JY, Chuang J, Manning CD, Ng AY, Potts C (2013) Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Stroudsburg PA. 1631-1642.

Stone RB, Kurtadikar R, Villanueva N, Arnold CB (2008) A customer needs motivated conceptual design methodology for product portfolio planning. *Journal of Engineering Design.*19(6):489-514.

Sullivan LP (1986) Quality function deployment. *Quality Progress.* 19(6), 39-50.

Tai KS, Socher R, Manning CD (2015) Improved semantic representations from tree-structured long short-term memory networks. *Proceedings of the 53st Annual Meeting on Association for Computational Linguistics.* Stroudsburg, PA. 1556-1566.

Tirunillai S, Tellis GJ (2014) Mining marketing meaning from online chatter: Strategic brand analysis of big data using Latent Dirichlet Allocation. *Journal of Marketing Research*. 51:463-479.

Tieleman T, Hinton G (2012) Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4.

Ulrich KT, Eppinger SD (2016) *Product design and development*, 6E. (New York, NY: McGraw-Hill).

Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. *Proceedings of the Conference On Human Language Technology and Empirical Methods in Natural Language Processing*. Vancouver BC. 347-354.

Wu H-H, Shich JI (2010) Applying repertory grids technique for knowledge elicitation in Quality Function Deployment. *Quality and Quantity.* 44:1139-1149.

Zahay D, Griffin A, Fredericks E (2004) Sources, uses, and forms of data in the new product development process. *Industrial Marketing Management*. 33:657-666.