A CLUSTERING AND WORD SIMILARITY BASED APPROACH FOR IDENTIFYING PRODUCT FEATURE WORDS

Suryadi, Dedy; Kim, Harrison
University of Illinois at Urbana-Champaign, United States of America

Abstract

Product designers need to capture feedback from customers in order to assess how the product performs and is perceived in the market. One such example of publicly available source of customer’s feedback is the online reviews in an e-commerce website. Two main difficulties in dealing with the reviews are finding relevant words related to a product and grouping different words that represent the same product feature. To overcome these difficulties, both lexical and distributional approaches are utilized in the paper. Using distributional information, words are embedded into real vector space using word2vec and then clustered. Using lexical information from WordNet, the head word for each cluster is identified by considering the similarity with the head words of other clusters. A comparison is made between using X-means and iterative c-means clustering with added word similarity information when breaking a cluster. In the case study of wearable technology products, starting from a large number of words, the approach is shown to identify relevant product feature words.

Keywords: Design informatics, Market implications, Case study

Contact:
Prof. Harrison Kim
University of Illinois at Urbana-Champaign
Industrial and Enterprise Systems Engineering
United States of America
hmkim@uiuc.edu

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 6: Design Information and Knowledge, Vancouver, Canada, 21.-25.08.2017.

1 INTRODUCTION

In product design it is important to capture the voice of consumer. It is suggested that linking the voice of customer to engineering, manufacturing, and R&D decisions leads to greater new-product success and more profitable products (Griffin and Hauser, 1993). To gather the voice of consumer, conventional methods such as survey or focus group study can be conducted, but it takes a considerable amount of resources. As an alternative, a publicly available source of data can be utilized, i.e. the online web page of a product that contains reviews written by consumers.

Online product reviews are important for both consumers and product designers. For consumers, it is reported that 68% online shoppers check at least four reviews before buying, and nearly 25% check at least eight reviews (Sun, 2012). For designers, it is a promising alternative to conventional survey techniques. In online product reviews, the reviewers communicate their opinions voluntarily, therefore a high level of authenticity can be expected (Decker and Trusov, 2010).

Once the reviews have been collected, however, there is a challenge in analyzing them. In general, the challenge lies in the nature of language, which can be ambiguous. For example, a word can have different meanings (e.g. "bank" can refer to a financial institution or a river margin), different part-of-speech (e.g. "bank" can be either noun or verb), different forms, etc. Furthermore, a sentence can be ambiguous to computer programs, although it is not for human (e.g. "She boarded the airplane with two suitcases." which is similar to "She boarded the airplane with two engines.") (Jackson and Moulinier, 2007). In addition, novel language, incomplete language, and errorful language are simply common (Bates et al., 1993).

In the particular case of analyzing product reviews, one of the challenges is automatically identifying product features that are discussed in the reviews. It is desirable to have a compact list of product features, i.e. different words that refer to the same product feature should be represented on the list by as few words as possible. The problem may then be stated as follows. Given the words from online reviews of a product, group the similar words and find a word to represent the group. The representatives of each group should reflect the relevant product (or service) features of the product.

This paper expands the previous works on analyzing online reviews (Suryadi and Kim, 2016a; Suryadi and Kim, 2016b). In those works, the identification was done manually based on subjective judgment. In order to have a method that does not rely on subjectivity, this paper focuses on improving the process such that it involves as little human supervision as possible but in an automated way.

The approach of proposed methods to solve the problem is based on embedding words into real vectors. The word2vec software from TensorFlow, which is based on Skip-gram model, is applied to do the embedding. Afterwards, clustering method is applied to group the words along with the information about tf-idf (term frequency - inverse document frequency) and similarity of the words. Thus, the paper's contribution is proposing a method that consists of word embedding and clustering procedures for finding product feature words from free format online reviews instead of a manual and subjective identification.

The rest of the paper is structured as follows. The next section presents literature reviews on previous related works and word embedding using Skip-gram model. It is followed by a section presenting the methods proposed in this paper, followed by the experimental data and results. Based on the results, a Discussion section is provided and finally the paper concludes with Conclusion and Future Works section.

2 LITERATURE REVIEW

2.1 Previous Works

There have been similar works published over the years. This section summarizes those works and highlights the differences with what is done in this paper. Archak et al. (2007) collects the frequent words and manually selects a subset of approximately 30 nouns as product feature words. Netzer et al. (2012) apply a rule-based approach to identify entity types (e.g. brands, products, names) for the collected noun phrases. Archak et al. (2011) use WordNet to cluster phrases into similar nouns and noun phrases, but do not offer the detail of the procedure.

Association rule mining is used to find frequent sets of words or a phrase that occurs together, which are called itemsets (Hu and Liu, 2004). Somprasertsri and Lalitrojwong (2008) extract product feature word candidates using linguistic filtering pattern. Abulaish (2009) identified feasible collection of product feature words using tf-idf (term frequency - inverse document frequency) approach. Tucker and Kim (2011) employ Bayesian Classification to determine the collected terms relating to the most frequent product feature words. Ma et al., (2013) apply Latent Dirichlet Allocation (LDA) and association mining to find frequent itemsets. There are, however, no further discussions about how to merge similar words into the same product feature word.

Guo et al., (2009) create virtual document for each candidate word and performs the LDA to group those. However, it uses the reviews that are already categorized into pros and cons, such that it starts with significantly less amount of words and, obviously, the words are much more inclined to be truly product feature words. Lee and Bradlow (2011) use the reviews with the same nature and state that they group the phrases according to similarity, which is measured as the cosine angular distance between word vectors. Furthermore, Zhai et al. (2011) rely on the assumption that from the beginning all product feature expressions have been identified.

Zhang et al. (2015) conclude that word2vec is suitable for text clustering in Chinese language. In their work, however, the task is finding synonyms rather than finding product feature words from scratch. The word2vec model is utilized to embed all words into real vectors and synonyms are then found by sorting the distance. In this paper, there are no initially predetermined words.

2.2 Word Embedding

The word2vec model is based on Skip-gram model that is a model architecture for learning distributed representation of words (Mikolov et al., 2013a). The architecture of Skip-gram model is shown in Figure 1.

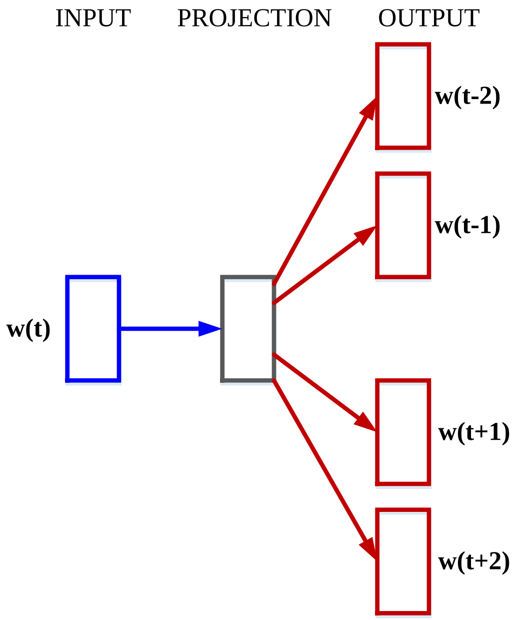

Figure 1. Skip-gram Model Architecture (Source: Mikolov et al., 2013a)

The model uses the current word w(t) as an input and predicts the neighboring words as the outputs. More specifically, it learns word representations to predict the neighboring words. In Figure 1, the neighboring words are two words before and after the current word.

Given the words w_1, w_2, …, w_T, as the inputs, the objective of the model is to maximize the log probability (Equation (1)) (Mikolov et al., 2013b):

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c,j\neq 0}\log p\left(w_{t+j}\middle|w_t\right) \tag{1}$$

The probability is defined based on the vector representations of the words, such that the probability of observing an output word w_{t+j} given an input word $w_t$is (Equation (2)) (Mikolov et al., 2013b):

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^{T} v_{w_I}\right)}{\sum_{w=1}^{W}\left({v'_{w}}^{T} v_{w_I}\right)} \tag{2}$$

where v_w and v'_w are the vector representations of a word, from Input to Projection and from Projection to Output, respectively, and W is the vocabulary size. Using gradient descent based method, the representations can be iteratively updated to minimize the objective defined in Equation 1. More detailed explanation of the learning process is written by Rong (2014).

The formulation in Equation 1 can be impractical because W is a large number. Therefore, word2vec uses the Negative Sampling approach. The approach modifies the objective as in Equation (3) (Mikolov et al., 2013b):

$$\log \sigma\left({v'_{w_O}}^T v_{w_I}\right) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_{n(w)}}\left[\log \sigma\left(-{v'_{w_i}}^T v_{w_I}\right)\right] \tag{3}$$

where σ is a logistic function, i.e. $\sigma(u) = 1/(1 + e^{-u})$, and $P_n(w)$ is a noise distribution. The objective aims to distinguish the target word w_O from words drawn from the noise distribution, where the number of words drawn is k for each data sample. More detailed explanation of Negative Sampling is given by Goldberg and Levy (2014).Given the nature of word embedding method that relies only on the textual data itself, it becomes an attractive approach for the task of finding the product feature words automatically. The proposed method therefore combines it with a clustering method, in order to group words with similar meaning into a reasonable number of groups.

3 METHODOLOGY

The proposed method consists of two stages, i.e. embedding words into real vector space and clustering the words. The first stage consists of pre-processing the textual data from the reviews and then utilizing word2vec to embed the words into real vectors. The output of the first stage, i.e. real vectors, becomes the input for the second stage. In the second stage, there are two clustering methods proposed, i.e. the straightforward X-means and the C-means with WordNet-based word similarity information.

The latter is proposed for two reasons. Firstly, C-means naturally outputs the membership of each data point to all centers, which can be subsequently useful to assign an ambiguous word to the words chosen as the centers (product feature words). In addition, word similarity is introduced because it is a lexical information that has not been considered in the first stage of word embedding. By introducing word similarity, it is expected that the output contains no product feature words that are too similar.

3.1 First Stage: Pre-processing & Word Embedding

The first stage is composed of these steps:

I. Pre-processing

1. Input: textual content of online consumer reviews.
2. Numbers, symbols, all single character alphabets are removed, except period.
3. Modals, prepositions, and "to be" verbs are removed, as well as the negative forms.

II. Word Embedding (word2vec)

4. All words on the pre-processed list are sorted based on frequency, starting from the highest.
5. Set vocabulary size = V. Collect the top (V-1) words from the list and let the remaining words be merged into an "UNK" (unknown) word.
6. Set the parameters in word2vec:
    a. batch size: the number of data for each batch in Stochastic Gradient Descent.
    b. embedding size: dimension of a word embedding vector.
    c. skip window: window of neighboring words from target word.
    d. number of skips: number of times a target word is used to generate a label.
    e. number of negative examples to sample.
7. Set the number of iterations, the initial word embedding vectors, as well as the parameters for the loss function and the learning parameter for Stochastic Gradient Descent.
8. In each iteration, Stochastic Gradient Descent is performed to improve the initial vectors.
9. Output: word embedding vectors for the top (V-1) words from the list.

3.2 Second Stage: Clustering

Once the embedding has been obtained from the first stage, either of the two procedures of clustering can be used, i.e. 1) X-means or 2) C-means and Word Similarity. The results of both procedures are compared and discussed in Section 4 and 5.

3.2.1 X-means Clustering

The steps for X-means clustering (Pelleg and Moore, 2000) is shown as follows.

1. Input: words, their respective embedding vector and tf-idf.
2. Further pre-processing: removing words that have no "noun" entry in their WordNet synsets.
3. Copy a word's vector as many as its tf-idf, rounded down to the nearest integer.
4. Initialization: k = 2, set initial center points.
5. Run k-means.
6. For each cluster:
    a. Compute the Bayesian Information Criterion (BIC) for the cluster.
    b. Create 2 new initial center points, run k-means for this cluster.
    c. Compute the BIC for the result with 2 centers.
    d. If BIC increases, then break the cluster into 2 with their respective centers.
    e. Repeat until $k > K_{max}$
7. Keep the clustering with k centers that gives the highest BIC score.
8. Compute the distance between all words to each of the k centers.
9. Output: the word closest to each center is collected as a feature word, without duplication.

3.2.2 C-means Clustering & Word Similarity

The steps for C-means and Word Similarity are shown as follows:

I. Initialization

1. Input: words and their respective word embedding vectors.
2. Further pre-processing: removing words that have no "noun" entry in their WordNet synsets.
3. Copy a word's vector as many as its tf-idf, rounded down to the nearest integer.
4. Initialization: k = 2, set initial center points = the vectors of 2 words with the highest tf-idf.

II. Performing C-means clustering

5. Run C-means.
6. Compute the distance between all words to each of the k centers. The word closest to each center is collected as the center words, without duplications.

III. Finding Initial Points for the Next Iteration

7. Construct hard clusters and find their respective centers.
8. For each hard cluster, evaluate if it should be broken into 2 clusters, i.e. compare the BIC estimate between clusters with 1 and 2 centers.
   If a 2-center cluster is better, then:
    a. Sort the words, starting from the farthest distance from the center. Denote the sorted words as w0, w1, w2, …
    b. The first initial point is w0.
    c. Compare the word similarity of w0 with w1 and w2. If w2 is more similar to w0, set w1 as the second initial point. Otherwise, move to w2 and w3, and so on.

   Otherwise, set the current center word as an initial point.
9. Check the number of initial points collected from step (8):
    a. If the number is the same with k and the words are the same with step (6), then it means all clusters remain the same. Stop the procedure here.
    b. Otherwise, if the number of initial points is still less than k, add the center words in step (5) that do not appear as center words in step (6) as initial points. Compare the BIC estimate between k and initial points; repeat c-means in step (4) if the latter is better, but stop otherwise.
10. Output: center words from the c-means in step (5).

4 EXPERIMENTAL DATA & RESULTS

4.1 Data Description

The online consumer reviews were collected from wearable technology product web pages in Amazon.com. In particular, the products are from the sub-category "Clips, Arm, Wristbands". There were 87 items (unique URLs) in this category, however some items share the same reviews, i.e. when the items only differ in color or shape. Without the duplicates, there are 59 items with unique reviews.

The reviews were collected from January 1st, 2015 until April 30th, 2016. The period is chosen such that the number of reviews are large enough. There were 72,880 reviews collected during the period. Wearable technology products are chosen as case study because they are relatively new to the market. They arrived less than a decade ago in the market. For example, Fitbit launched its first tracker in 2008, while Jawbone's tracker was initially launched in 2011. From the design perspective, it still provides a great room to improve the products based on customers' feedback. Moreover, it is reasonable to assume that people would seek and share information about the products, due to their newness.

Three examples of the collected reviews from a product called "Mio FUSE Heart Rate, Sleep + Activity Tracker", with star ratings assigned by consumer are 1, 3, and 5 stars, respectively. It can be seen from the examples on Table 1, the reviews are all written in free format, i.e. they do not come in the format of pros and cons sections.

Table 1. Examples of Reviews

Terribly inaccurate. Review on PC magazine indicated best all around activity/heart monitor, especially for weight training. NOT true. While a range of 3-5 BPM is acceptable, a variation of 20 BPM is not. Waste of money. I'm back to the Polar product with a chest strap/ heart sensor for accuracy.
I bought this primarily to use it for tracking swim workouts. Sadly, I was never able to get a good result in the water. While swimming it would report a heart rate, but it was very erratic and rarely correct. At one point it had my heart rate at 36 bpm. I tried moving it up my arm, wearing it tighter, wearing it on the inside of my arm, and on the other arm. Nothing helped.
Great way to track your pulse while working out - connects with an app on your phone easily, very intuitive.

4.2 Results

After pre-processing the reviews, there are 20,613 unique tokens found. The words are sorted based on frequency and the vocabulary size is set as 5,000. It means that the top 4,999 words are kept and the remaining are merged into the "UNK" word. The top and bottom 20 words from the vocabulary are shown on Table 2, with the frequency of each word is shown as a number inside the parentheses.

Table 2. Top and Bottom 20 Words from the Vocabulary

Top 20 Words	Bottom 20 Words
it (139949)	outcome (14)
and (101959)	mount (14)
to (101075)	components (14)
my (64986)	guided (14)
UNK (46696)	blog (14)
this (38510)	oddly (14)
that (34207)	strictly (14)
with (30285)	hmmm (14)
have (29475)	disable (14)
not (26963)	shade (14)
fitbit (26446)	reconditioned (14)
but (26113)	recovered (14)
you (23999)	pj (14)
love (22369)	instructor (14)
me (20489)	confirms (14)
so (19987)	summaries (14)
one (17695)	advocate (14)
had (15918)	cloth (14)
up (15866)	filed (14)
great (15481)	wishes (14)

The parameters for word2vec in the experiment are set as follows:

a. batch size: 256
b. embedding size: 128
c. skip window: 1
d. number of skips: 2
e. number of negative examples to sample: 64

For the Stochastic Gradient Descent in word2vec, the initial word embedding vectors are real numbers chosen randomly from a continuous uniform distribution between [-1, 1]. The learning parameter is set to be 1.0. It is run as many as 100,001 iterations. The output of word2vec is all words embedded into vectors with 128 dimensions of real values.

Afterwards, both in X-means and C-means, the words are weighted by their tf-idf values. A list of Top 20 words with highest tf-idf and their respective tf-idf values inside the parentheses are: love (2194.37), awesome (1121.05), product (1108.80), fitbit (972.40), excellent (793.77), perfect (597.69), charge (576.22), use (573.67), device (492.35), battery (485.12), gift (467.17), hr (455.86), band (454.57), day (445.64), wife (427.09), tracker (426.99), accurate (419.54), track (414.00), sleep (409.52), work (406.77).

Based on a particular embedding output from word2vec, there are 20 unique product feature words obtained as the output of X-means procedure. The output of X-means is always the same as long as the word embedding vectors do not change. Meanwhile, there are 31 unique product feature words obtained as the output of running C-means and Word Similarity procedure 20 times. The latter procedure utilizes the BIC estimate, such that the result of each run can be different. The outputs of both procedures are shown on Table 3.

Table 3. Product Feature Words Obtained from Two Different Procedures

X-means		C-means and Word Similarity	
*band	*purchase	activity	*product
clip	rubber	*band	*purchase
cover	screen	charge	sleep
*device	strap	concept	time
flex	thing	-day	track
gadget	*tracker	*device	*tracker
*hr	*use	gauge	unit
*-love	*watch	-gift	upgrade
-mine	wristband	heart	*use
*product	*zip	-hide	*watch
		*hr	wear
		-increase	-week
		item	-wife
		*-love	wrist
		monitor	*zip
		-move	

Note:

*: found in both procedures

-: not a reasonable product/service feature for wearable technology products

The results are visualized in Figure 2 and Figure 3. Since it is impossible to visualize the 128-dimension word embedding vectors, the dimension is reduced by means of Principal Component Analysis (PCA) into 2-dimension vectors. Each word is represented by a circle whose size is proportional to the word's tf-idf value, while the centers are represented by diamond shapes.

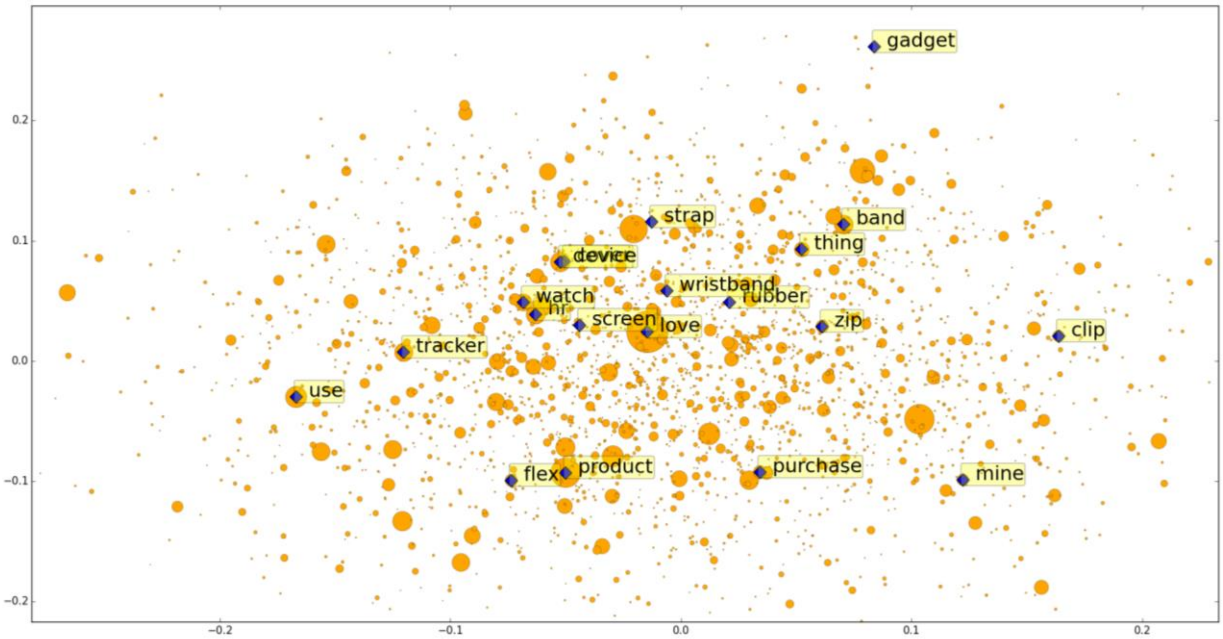

Figure 2. X-means Result

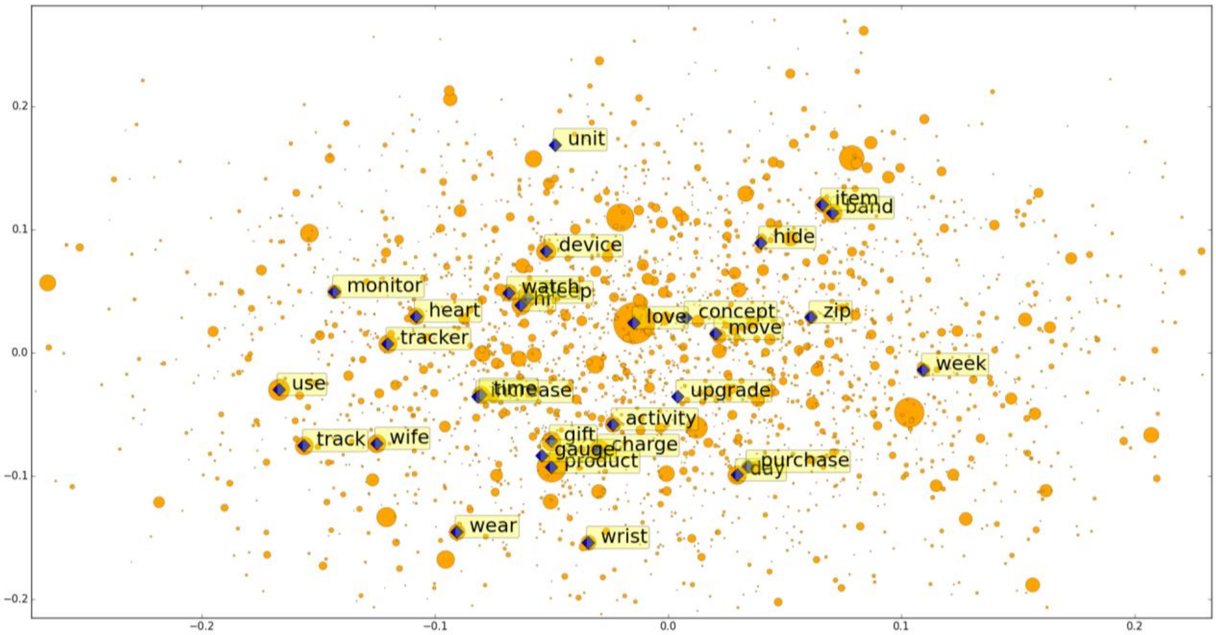

Figure 3. C-means Result (from 20 Runs)

5 DISCUSSION

There are 20 product feature words obtained from X-means. Two of those are considered unrelated to any product/service feature of wearable technology products, i.e. "love" and "mine". Therefore, 18/20 (90%) words are considered reasonable. Among the 18 words, some words may share similar meaning, such as "device", "gadget", "product", "thing", and "tracker". However, the plot in Figure 2 shows that according to the distributional information from the reviews, those words are not close to each other and therefore X-means procedure assigns those words as their respective cluster centers.

In C-means and word similarity procedure, each time a cluster is considered to be broken into 2 clusters, the initial points for the new clusters are chosen with specific consideration. It is that after sorting the words starting from the farthest word from the center, the first initial point is chosen as the farthest point from the current center. The second initial point is then chosen by comparing the j and (j+1) words' with the first initial point. If the word (j+1) words is less similar, then the comparison continue with the (j+1) and (j+2) words, and so on. Otherwise, the word j is picked as the second initial point. This is done in attempt to create initial centers that are as dissimilar as possible to each other, because it is known that C-means (which is based on K-means) depends on initial points. The similarity metric used in the paper is Jiang-Conrath similarity, which gives the similarity in the range of [0, 1].

There are 31 product feature words from the second procedure. There are 8 of them are considered unrelated to any product/service feature, i.e. "day", "gift", "hide", "increase", "love", "move", "week", and "wife". Therefore, 23/31 (74%) words are considered reasonable. Although similarity consideration has been imposed while finding the initial points in C-means, the similar words such as "product", "device", and "tracker" still appear together all the time. However, there are three interesting words captured by this procedure are "charge", "concept", and "sleep". The word "charge" can refer to either the process of charging battery or the name of a product, i.e. Fitbit Charge. The word "concept" can refer to the design of a product and the last refers to the function related to sleep tracking of the products.

Among the 10 words that are obtained by both procedures, there is a word unrelated to any product/service feature, i.e. "love". The word reasonably appears as cluster center because it has the highest tf-idf value and significantly higher than other words, i.e. nearly two times of the second highest. To justify the usage of tf-idf to weigh words, there should be a way to weigh words differently among the 5,000 words in the selected vocabulary. An initial experiment without using any weight results in a single cluster for all words. The word "love" understandably has very high tf-idf as well, because it tends to appear a lot on the positive reviews (high term frequency), yet not nearly all reviews are positive (high inverse document frequency).

The X-means clustering method is mainly chosen to match the nature of the problem. In the problem, there are no quantifiable ways to predeterminedly decide the number of features in a product. X-means clustering works by running K-means clustering while iteratively breaking clusters into smaller ones based on the Bayesian Information Criterion (BIC) value. Thus, in the end it automatically provides the number of clusters as well.

The C-means clustering method allows a data point to have membership in more than one cluster. It aligns with the fact that a word can have different degree of similarity with other words. Therefore, a method is also proposed for a C-means approach, complemented by word similarity information.

There is a limitation for the proposed methods in capturing the voice of customers. The method utilizes tf-idf that requires the count of words. However, there are occurrences of product feature words that are not properly counted due to reference problems. For example, it is common in the reviews that customers refer to the previously mentioned feature simply by the word "it". Therefore, it affects the precision of the obtained product feature words, as well as the clustering of words into the product feature words.

6 CONCLUSION & FUTURE WORKS

The paper proposes procedures to automatically find product feature words from free format online consumer reviews. It starts with an enormous number of words, i.e. more than 20,000 and ends with 20 words (X-means) and 31 words (C-means and Word Similarity). Most of the words are found to be reasonably related to product features. X-means gives less unrelated words, while C-means and word similarity output more diverse words, such as "charge", "concept", and "sleep".

The finding enables the assignment of thousands of words into a group of relevant product feature words discussed in the reviews. For example, using the clustering result, there is an objective and quantifiable way to assign a word into the most possible product feature, e.g. "alarm" into "watch". It contributes greatly to the next step, i.e. analyzing the sentiments that people express towards the product features and how it is related to the sales of the product. Therefore, product designers are able to analyze product features that matter for consumers and focus to improve those features.

Finally, there are some future works to improve the result. First, a better way to impose word similarity into the procedure is required to reduce the words with similar meaning, for example "device", "product" and "tracker". Second, the data analysis will go deeper into filtering out the fake reviews, i.e. the ones that are not written by customers who truly purchase the product with his/her own money. Although fake reviews mostly affect sentiment analysis, it also introduces a bias into the clustering-based approach that is used in this paper.

REFERENCES

Abulaish M., Jahiruddin, Doja M.N., and Ahmad T. (2009), “Feature and opinion mining for customer review summarization”, *Pattern Recognition and Machine Intelligence. PReMI*, Vol. 5909. http://dx.doi.org/10.1007/978-3-642-11164-8_35

Archak, N., Ghose, A., and Ipeirotis, P. G. (2007), “Show me the money! Deriving the pricing power of product features by mining consumer reviews”, *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '07*, pp. 56–65. https://doi.org/10.1145/1281192.1281202

Archak, N., Ghose, A., and Ipeirotis, P. G. (2011), “Deriving the pricing power of product features by mining consumer reviews”, *Management Science*, Vol. 57(8), pp. 1485–1509. http://dx.doi.org/10.1287/mnsc.1110.1370

Bates, M., Bobrow, R. J., and Weischedel, R. M. (1993), “Challenges in natural language processing”, *Studies in Natural Language Processing*, Cambridge University Press, USA.

Decker, R., and Trusov, M. (2010), “Estimating aggregate consumer preferences from online product reviews”, *International Journal of Research in Marketing*, Vol. 27, pp. 293–307. http://dx.doi.org/10.1016/j.ijresmar.2010.09.001

Goldberg, Y., and Levy, O. (2014), “word2vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method”, *CoRR*, Available at: http://arxiv.org/abs/1402.3722 (12/18/2016).

Guo, H., Zhu, H., Guo, Z., Zhang, X., and Su, Z. (2009), “Product feature categorization with multilevel semantic association”, *Proceeding of the 18th ACM Conference on Information and Knowledge Management (CIKM '09)*, pp. 1087–1096. http://doi.acm.org/10.1145/1645953.1646091

Hu, M., and Liu, B. (2004), “Mining opinion features in customer reviews”, *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, pp. 168–177. http://doi.acm.org/10.1145/1014052.1014073

Jackson, P., and Moulinier, I. (2007), “Natural language processing for online applications: Text retrieval, extraction and categorization”, 2nd rev. ed., John Benjamins Publishing Company, USA.

Lee, T., and Bradlow, E. (2011), “Automated marketing research using online customer reviews”, *Journal of Marketing Research*, Vol. 48(5), pp. 881–894.

Ma, B., Zhang, D., Yan, Z., and Kim, T. (2013), “An LDA and synonym lexicon based approach to product feature extraction from online consumer product reviews”, *Journal of Electronic Commerce Research*, Vol. 14, No. 4, pp. 304–314.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013a), “Efficient estimation of word representations in vector space”, *International Conference on Learning Representations*.

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

Netzer, O., Feldman, R., Goldenberg, J., and Fresko, M. (2012), “Mine your own business: market-structure surveillance through text mining”, *Marketing Science*, Vol. 31, No. 3, pp. 521–543. http://dx.doi.org/10.1287/mksc.1120.0713

Pelleg, D. and Moore, A. (2000), “X-means: extending K-means with efficient estimation of the number of clusters”, *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 727–734.

Rong, X. (2014), “word2vec parameter learning explained”, *CoRR*, Available at: http://arxiv.org/abs/1411.2738 (12/18/2016).

Somprasertsri, G., and Lalitrojwong, P. (2008), “Automatic product feature extraction from online product reviews using maximum entropy with lexical and syntactic features”, *IEEE International Conference on Information Reuse and Integration*, pp. 250–255.

Sun, M. (2012), “How does the variance of product ratings matter?”, *Management Science*, Vol. 58(4), pp. 696–707. http://dx.doi.org/10.1287/mnsc.1110.1458

Suryadi, D., and Kim, H. M. (2016a), “Identifying sentiment-dependent product features from online reviews”, *Design Computing and Cognition '16*, pp. 721–740.

Suryadi, D., and Kim, H. M. (2016b), “Identifying the relations between product features and sales rank from online reviews”, *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*.

Tucker, C., and Kim, H. (2011), “Predicting emerging product design trend by mining publicly available customer review data”, *International Conference on Engineering Design*.

Zhai, Z., Liu, B., Xu, H., and Jia, P. (2011), “Clustering product features for opinion mining”, *ACM International Conference on Web Search and Data Mining*.

Zhang, D., Xu, H., Su, Z. and Xu, Y. (2015), “Chinese comments sentiment classification based on word2vec and SVMperf”, *Expert Systems with Applications*, Vol. 42, pp. 1857–1863. http://dx.doi.org/10.1016/j.eswa.2014.09.011