AUTOMATIC IDENTIFICATION OF PRODUCT USAGE CONTEXTS FROM ONLINE CUSTOMER REVIEWS

Suryadi, Dedy; Kim, Harrison

University of Illinois at Urbana-Champaign

ABSTRACT

There are three product design contexts that may significantly affect the design of a product and customer preferences towards product attributes, i.e. customer context, market context, and usage context factors. The conventional methods to gather product usage contexts may be costly and time consuming to conduct. As an alternative, this paper aims to automatically identify product usage contexts from publicly available online customer reviews. The proposed methodology consists of Preprocessing, Word Embedding, and Usage Context Clustering stages. The methodology is applied to identify usage contexts from laptop customer reviews, which results in 16 clusters of usage contexts. Furthermore, analyzing the review sentences explains the separation of "playing games" –which is more related to casual gaming, and "gaming rig" –which implies high computing power requirements. Finally, comparing customer review with manufacturer's product description may reveal a discrepancy to be investigated further by product designer, e.g. a customer suggests a laptop for basic use, although the manufacturer's description describes it for heavy use.

Keywords: Design informatics, User centred design, Crowdsourcing

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

Cite this article: Suryadi, D., Kim, H. (2019) 'Automatic Identification of Product Usage Contexts from Online Customer Reviews', in *Proceedings of the 22nd International Conference on Engineering Design (ICED19),* Delft, The Netherlands, 5-8 August 2019. DOI:10.1017/dsi.2019.257

1 INTRODUCTION

Product usage context is one of the product design contexts that may significantly affect customer's preference for product attributes (Green *et al.*, 2005). In other word, different usage situation or environment may shift customer's preference from one product attribute to the other. Green *et al.* (2004) specifically lists three benefits of understanding product usage context, i.e. (1) improved customer needs gathering, (2) improved setting of target design values, and (3) leveraging the known to design for the unknown. Therefore, understanding product usage context may finally lead to successful products that satisfy customer needs. Furthermore, identifying product usage contexts may benefit other applications as well, such as choice modeling (He *et al.*, 2012) and latent customer needs elicitation (Zhou *et al.*, 2015).

The conventional methods to gather product usage contexts are one-on-one interviews, focus groups, and observations of customers using an existing product to perform particular tasks (Ulrich and Eppinger, 2004). However, those methods may be costly and time consuming to conduct. As the alternative, online customer reviews are publicly and readily available to analyze. In this paper, most of the reviews are assumed to be authentic, since customers voluntarily invest time and energy to share their opinions (Decker and Trusov, 2010).

A previous research by Zhou *et al*. (2015) identified usage contexts from online customer reviews of a Kindle tablet. Based on the identified usage contexts, the contexts that have small proportions are considered extraordinary and the corresponding customer needs are thus considered latent. However, all usage contexts are predefined, e.g. "*On a Trip*" under Contextual Events category in Figure 1.

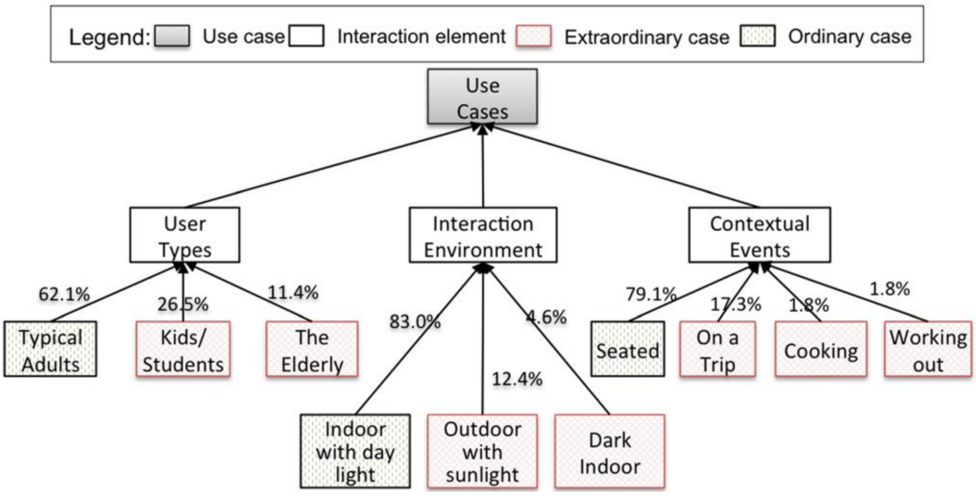

Figure 1. Use cases extracted from online product reviews (Source: Zhou et al. (2015))

The disadvantages of predefining usage contexts as in Zhou *et al*. (2015) are:

- The actual review data may not agree with the predefined groups, e.g. "*Kids/Students*" under User Types category may contain greatly varied contexts due to the fact that the group may range from elementary school kids to graduate students.
- Most importantly, to predefine a reasonable set of usage contexts, it is imperative to read the reviews. Since the review volume is usually massive, this may be time consuming and thus diminishes the advantage of utilizing online reviews compared to the conventional methods.

Therefore, to overcome the disadvantages of the previous work, the purpose of this paper is to automatically identify product usage contexts from online customer reviews. The main tools to be used in the methodology are word embedding and clustering. Once the clusters of usage contexts have been obtained, this paper contributes to enable the comparison between customer's usage context and designer's intended usage context. The comparison shows whether or not customers use the product in the context that is intended by the designer.

The rest of the paper is organized as follows. Section 2 reviews relevant literature related to product design contexts and word embedding technique. Section 3 presents the proposed methodology to identify product usage contexts from online customer reviews automatically. Section 4 applies the proposed methodology to a case study. Section 5 discusses the method and metric selection, as well as the results from the case study. Finally, Section 6 concludes the paper.

https://doi.org/10.1017/dsi.2019.257 Published online by Cambridge University Press

2 LITERATURE REVIEW

2.1 Product design contexts

There are three product design contexts that may significantly affect the design of a product (Green *et al.*, 2004), i.e. customer context factors, market context factors, and usage context factors. Not only affecting the design, those factors influence customer preferences as well (Green *et al.*, 2006). Customer context factors are related to the attributes of a customer, e.g. education level, wealth, values, beliefs. Market context factors are related to the attributes of competitor products in the market, e.g. cost and features of competitor products. Usage context factors, the ones that receive the least attention from textbook methodologies, are related to the applications of a product, as well as the relevant situations accompanying the applications.

Green *et al*. (2006) consequently divide the usage context factor into two, i.e. application context ("*how*") and environment context ("*where*"). Similarly, LaFleur (1992) includes application environment as one of the variables that contribute to total product realization and defines it as the actual situation a product will encounter, including the actual tasks to perform. Even earlier, Belk (1975) has proposed Task Definition as one of the five groups of situational characteristics that influence consumer behavior. It is defined as a situation that includes an intent or requirement to select, shop for, or obtain information about a general or specific purchase. These literatures confirm the importance of identifying product usage contexts.

2.2 Word embedding models

Mikolov *et al*. (2013a) introduce a word embedding technique to learn high-quality word vectors from data sets with millions of words in the vocabulary. The learning process is performed through a neural network that consists of 3 layers, i.e. Input, Projection, and Output. Two architectures of network are introduced, i.e. Continuous Bag-of-Words (CBOW) and Continuous Skip-gram models, which are shown in Figure 2.

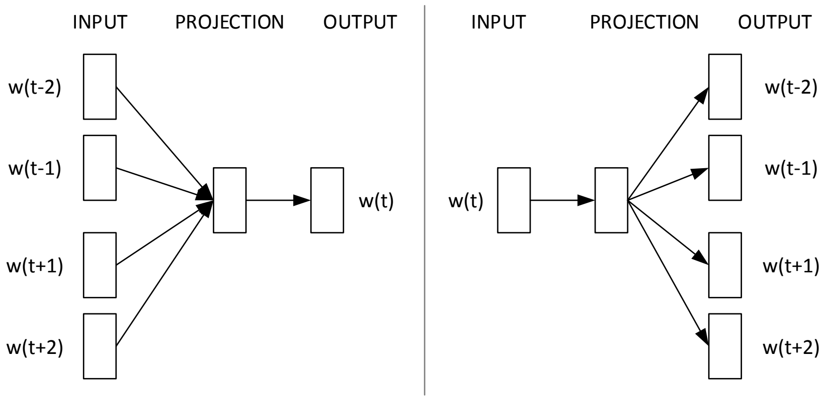

Figure 2. Network architecture: CBOW (left) and Skip-gram (right) (Source: Mikolov et al. (2013a))

The learning process involves analyzing words in a sentence through a moving window of words. In the example shown in Figure 2, the window is 2 words. At the *t*-th word in a sentence (*w(t)*), the word is called a target word. Meanwhile, 2 words before and 2 words after *w(t)* are called context words. In CBOW, the model predicts the target word, given context works. In Skip-gram, it is the opposite, and the probability formula is shown in Equation (1) (Mikolov *et al*., 2013b).

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

where:

$p(w_O \mid w_I)$ = probability of observing w_O as the target word, given w_I as the context word

v_w = vector representation of a word w in the Input layer

v'_w = vector representation of a word w in the Projection layer

https://doi.org/10.1017/dsi.2019.257 Published online by Cambridge University Press

Starting with random vectors to represent words, the model updates the vectors with respect to the objective of maximizing the average likelihood of observing the actual target words (CBOW) or context words (Skip-gram) over all windows, as shown in Equation (2) (Mikolov *et al*., 2013b). The detailed explanation of backpropagation procedures to update the word vectors are presented in Rong (2014).

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j} \mid w_t) \tag{2}$$

where:
c = the size of the window of words
T = cardinality of words in the corpus

3 METHODOLOGY

The proposed methodology to automatically obtain and cluster usage contexts from online customer reviews consists of three major stages, i.e. Preprocessing, Word Embedding, and Usage Context Clustering. The methodology is summarized in Figure 3.

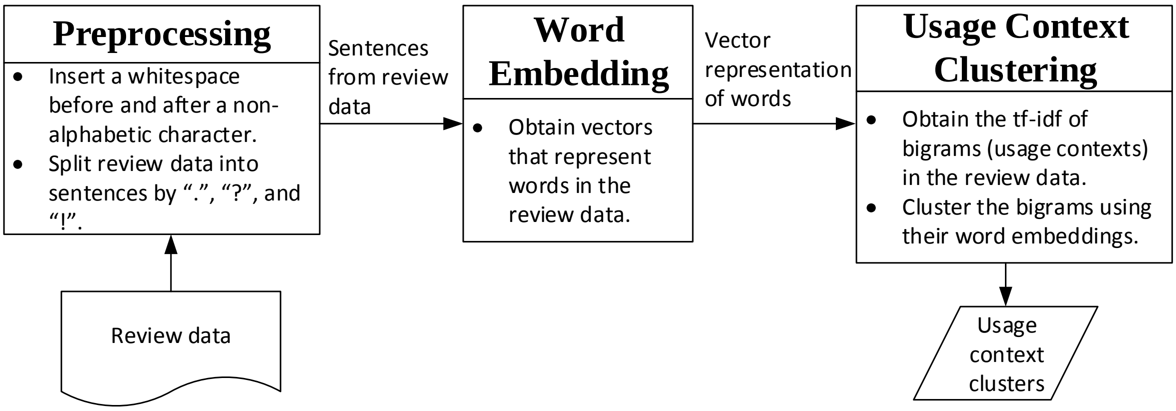

Figure 3. Proposed methodology

In the Preprocessing stage, the input is online customer review data. Firstly, for each non-alphabetical character, a whitespace is inserted before and after the character. This step is performed because the next stage, i.e. Word Embedding, assumes that words in a sentence are separated by whitespaces. The importance of this step may be shown by analyzing the following sentence, i.e. "**Mobile office app is free on this device,word excel, PowerPoint ,also onenote,that is perfect ,great bouns.*". Without this step, "**Mobile*", "*device,word*", "*onenote,that*", and "*,great*" would be assumed to be valid words (unigrams) and, consequently, affect the quality of the word embedding. After inserting the whitespaces, each customer review are finally split into sentences based on the occurrence of a dot, a question mark, or an exclamation mark.

In the Word Embedding stage, the input is a set of sentences from online customer review data. A Python package called *gensim* (Řehůřek and Sojka, 2011) is used to implement the word embedding. There are several parameters to be determined for the word embedding. *Size* parameter determines the dimension of a vector that represents a word. *Window* parameter determines the number of context words that are adjacent to the target word to be included in learning the prediction of the target word. Finally, since words that appear with very low frequency do not contribute much in learning the relationship between adjacent words, *min_count* parameter determines the minimum frequency of a word to be included in the vocabulary. The output of this stage is the representation of words in real vectors.

In the Usage Context Clustering stage, the usage contexts are collected based on the assumption that a usage context is a bigram (a pair of words). This assumption is justified by ranking the bigrams based on tf-idf (term frequency, inverse document frequency) metric and showing that the top bigrams reasonably capture the usage contexts.

These collected bigrams are then clustered to group the bigrams with similar meaning. To cluster the bigrams, it is important to consider both words in the bigram. For example, in the context of laptops, "*web browsing*" and "*internet browsing*" are similar activities, but "*playing games*" and "*playing music*" are quite different despite having the same verb. Thus, in order to incorporate both words in the bigram, each bigram is represented by the addition of its components (unigrams), e.g. "*web*

https://doi.org/10.1017/dsi.2019.257 Published online by Cambridge University Press

browsing" is represented by a vector that is obtained by adding the vector of "*web*" and the vector of "*browsing*".
Once the vector representations of bigrams have been obtained, two clustering methods are implemented, i.e. x-means clustering and spherical k-means clustering. Pelleg and Moore (2000) proposes x-means clustering method. It starts with the lower bound of K (number of clusters) and performs k-means clustering. Then, the cluster centers are split and k-means is performed. The clustering result after splitting is evaluated based on the Bayesian Information Criterion (BIC) metric. If the splits improve the BIC value, the splits are made permanent and the algorithm continues to the next split. Otherwise, the split of cluster centers is canceled and the algorithm stops. In spherical k-means clustering, the algorithm is the same with regular k-means; except that the distance metric is based on cosine similarity, instead of Euclidean.

4 CASE STUDY

The proposed methodology is applied to a set of customer reviews for laptops in Amazon.com website. The data was collected on December, 2017 for the laptops in Traditional Laptops category. There were 263,731 customer reviews, with the latest review was posted on December 13th, 2017. For the case study of this paper, the data was filtered by excluding laptops whose names contain the following words: "*chrome*", "*detach*", "*tablet*", "*studio*", and "*surface*". The laptops with those names are considered to have specifically different usage contexts compared to a conventional laptop. As the result, 218,570 reviews from 5,419 laptops with unique Amazon Standard Identification Numbers (ASINs) remain in the data set.
An example of a customer review is shown in Figure 4 for a laptop named "*Dell Latitude D630 + Windows 7 notebook laptop computer*". From the review, it may be implied that the customer uses the laptop mostly for basic activities, such as playing games and browsing the internet. In Section 5, the comparison between this customer review and the manufacturer's description of the laptop will be discussed.

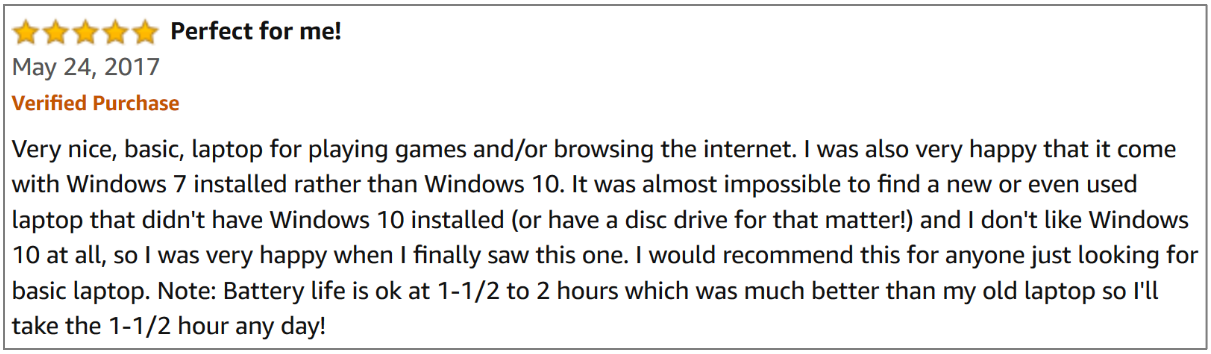
Perfect for me!
May 24, 2017
Verified Purchase
Very nice, basic, laptop for playing games and/or browsing the internet. I was also very happy that it come with Windows 7 installed rather than Windows 10. It was almost impossible to find a new or even used laptop that didn't have Windows 10 installed (or have a disc drive for that matter!) and I don't like Windows 10 at all, so I was very happy when I finally saw this one. I would recommend this for anyone just looking for basic laptop. Note: Battery life is ok at 1-1/2 to 2 hours which was much better than my old laptop so I'll take the 1-1/2 hour any day!

Figure 4. Example of a customer review in the data set (Source: https://www.amazon.com/gp/customer-reviews/R3CTBOL74I827W?ASIN=B004VC3NM0, accessed date: November 21, 2018)

4.1 Results of word embedding stage

After the preprocessing step, the sentences from the customer reviews become the input for word embedding. In this paper, the word embedding is implemented using *word2vec* module from a Python package called *gensim* (Řehůřek and Sojka, 2011). The parameters for word embedding are set as follows: size = 50, window = 3, min_count = 5, and other parameters are left as defaults from *gensim*. Since the word embedding parameter optimization is a goal of this paper, it is outside of the paper's scope and, thus, it is not performed here.
In order to show the word embedding performance, Table 1 shows the cosine distances between vectors of selected words, i.e. "*browsing*", "*surfing*", "*writing*", and "*typing*". On the table, the distance between a word and itself is omitted. It may be observed that "*browsing*" has the smallest cosine distance to "*surfing*", while having cosine distance above 0.5 to the other two words, i.e. "*writing*", and "*typing*". On the other hand, "*writing*" is closest to "*typing*", among all of the selected words. Based on this small example, it is shown that, to some extent, word embedding is successfully capture the similarity of words into vector of real numbers.

https://doi.org/10.1017/dsi.2019.257 Published online by Cambridge University Press

Table 1. Cosine similarity distance between selected words

	surfing	writing	typing
browsing	**0.05242**	0.54467	0.63396
surfing		0.54891	0.69206
writing	0.54891		**0.32669**

At this point, the word vectors have been obtained. In the next step, a set of reasonable usage contexts needs to be collected. In this paper, the usage contexts are assumed to be in form of bigrams, i.e. pairs of words. The bigrams are collected using a Python package called *sklearn* (Pedregosa *et al*., 2011). The collected bigrams are filtered, such that only bigrams that contain a word that ends with "*-ing*" are retained. The important bigrams, based on the tf-idf (term frequency, inverse document frequency) metric, are shown in Table 2.

Table 2. Top 12 bigrams with the highest tf-idf

Bigram	tf-idf
operating system	0.0005204
gaming laptop	0.0004915
stopped working	0.0004801
web browsing	0.0003548
word processing	0.0002301
video editing	0.0002037
web surfing	0.0001780
viewing angles	0.0001765
watching movies	0.0001476
running windows	0.0001464
playing games	0.0001315
internet browsing	0.0001249

While the bigrams in Table 2 are reasonable to describe usage contexts, tf-idf metric is unable to either group the bigrams with similar meanings, e.g. "*web browsing*" and "*internet browsing*", or separate the bigrams with different meanings, e.g. "*playing games*" and "*playing music*". Therefore, the final step would be clustering the bigrams into cluster of usage contexts.

4.2 Results of usage context clustering stage

In order to cluster the bigrams, two clustering methods are applied, i.e. x-means and spherical k-means. The comparison between the results from those methods are shown in Table 3. It can be seen that both methods outputs similar number of clusters, i.e. 16 (x-means) and 17 (spherical k-means). Furthermore, 10 bigrams are discovered to be the most frequent bigram in their clusters for both methods. Finally, based on two distance-based quantitative metrics, the result from x-means clustering is determined to be the better one. The reasoning behind the metrics and the qualitative assessment of the clustering results will be discussed further in Section 5.

Once the clusters have been obtained, the proportion of each usage context cluster may be computed as well. The proportions are represented in a pie chart shown in Figure 5, in which each cluster is labeled by the most frequent bigram in the cluster. It can be seen that the top 5 usage contexts obtained from the customer reviews are "*stopped working*", "*word processing*", "*operating system*", "*web browsing*", and "*viewing angles*".

https://doi.org/10.1017/dsi.2019.257 Published online by Cambridge University Press

Table 3. Comparison of clustering results

	x-means Clustering	Spherical k-means Clustering
Number of clusters	16	17
Most-frequent bigram in each cluster (the bigrams that appear only in one of the two clustering methods' results is highlighted in bold)	**docking station** finger scrolling **gaming rig** learning curve operating system playing games **processes running** processing power selling point star rating stopped working viewing angles **watching movies** web browsing **word processing** **writing papers**	**cooling system** finger scrolling **hours depending** learning curve **living room** operating system playing games processing power **processing speed** selling point **shipping label** star rating stopped working **updating drivers** **video editing** viewing angles web browsing
Average cosine distance between most-frequent bigrams	0.9056	0.9399
Average cosine distance of bigrams within a cluster	0.5009	0.5968

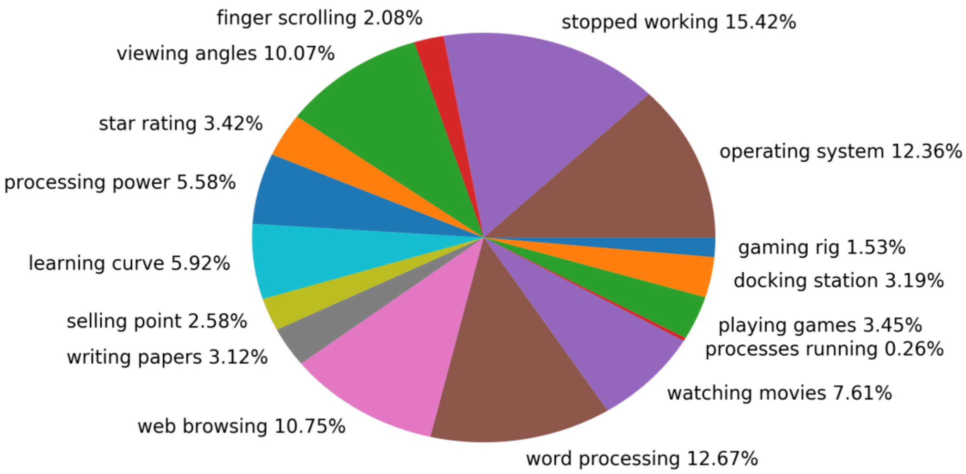

Figure 5. Proportion of usage contexts

The clustering result may also be assessed qualitatively by examining the bigrams that comprise each cluster. In order to remove relatively improbable bigrams, the Phrase module from *gensim* package is implemented. The module, based on Pointwise Mutual Information, outputs a vocabulary of bigrams that are likely to be phrases. As the result after filtering, for each cluster, the top 5 most frequent bigrams are listed in Table 4; with the exception of the "*watching movies*" cluster that shows the top 6 bigrams. Using the previous example, it may be seen that "*web browsing*" and "*internet browsing*" are in the same cluster. On the other hand, "*playing games*" and "*playing music*" are not, due to "*playing music*" being clustered into "*watching movies*". Further discussion about the qualitative assessment of the clustering result will be presented in Section 5.

https://doi.org/10.1017/dsi.2019.257 Published online by Cambridge University Press

Table 4. Most frequent bigrams in each usage context cluster

docking station	**finger scrolling**	**gaming rig**	**learning curve**
living room	typing papers	computing power	people complaining
computing needs	clicking noise	hardcore gaming	manufacturing defect
carrying case	tracking pad	gaming sessions	researching laptops
engineering student	cursor jumping	duty computing	mind blowing
operating system	**playing games**	**processes running**	**processing power**
operating systems	demanding games	loading webpages	charging port
cooling system	demanding game	handles multitasking	power saving
updating drivers	streaming media	spring loaded	stopped charging
transferring files	playing minecraft	resource hogging	charging cable
selling point	**star rating**	**stopped working**	**viewing angles**
consider buying	hours depending	stop working	viewing angle
suggest buying	restocking fee	stops working	cooling pad
considering returning	time consuming	quit working	cooling fan
regret purchasing	contacting asus	saving mode	plastic casing
watching movies	**web browsing**	**word processing**	**writing papers**
watching videos	web surfing	video editing	checking email
watching netflix	internet browsing	photo editing	reading reviews
watching youtube	internet surfing	processing speed	writing documents
streaming movies	browsing internet	streaming video	checking emails
playing music			

5 DISCUSSION

In clustering the usage contexts (bigrams), two different clustering methods are implemented. X-means clustering is chosen because it has the ability to determine the number of clusters using Bayesian Information Criterion (BIC) metric, which has taken both the likelihood of the data under a particular clustering and the number of clusters into account. Spherical k-means is chosen due to its usage of cosine similarity as the basis of the distance metric, which may capture the similarity between word vectors better than a Euclidean-based metric. In order to determine the number of clusters (*k*), a metric in Equation (3) is proposed. As the *k* goes larger, the inertia becomes smaller, but the sum of distance between the most frequent bigrams in each cluster gets larger; and thus the *k* that gives the minimum value may be obtained. In this paper's case study, the *k* is found to be 17.

$$\frac{I}{k} + \sum_{c1=1}^{k} \sum_{c2=c1+1}^{k} dist\ \ v_{c1} + v_{c2} \tag{3}$$

where:

$dist\ \ v_{c1} + v_{c2}$ = cosine distance between word vectors v_{c1} and v_{c2}

I = inertia (within sum-of-squares) of the clustering

k = number of clusters

v_{c1}= word vector of the most frequent bigram in cluster c1

The two clustering results are compared using two metrics. The first metric is average cosine distance between most-frequent bigrams. Bigrams in different clusters should be different enough, but they should all come from the same domain, e.g. laptop usage contexts; and thus should not involve distantly related phrases, e.g. shipping label. Therefore, the first metric is used to measure the cohesiveness of the set of most frequent bigrams in the clusters. The second metric is average cosine distance of bigrams within a cluster, which simply measures the cohesiveness of bigrams within clusters. A good clustering should create clusters of bigrams with cohesive meaning. Based on these two smaller-the-better metrics, the result of x-means clustering is the better one for the case study.

https://doi.org/10.1017/dsi.2019.257 Published online by Cambridge University Press

The usage context clustering in Table 4 shows that the words in a cluster are relatively similar in either meaning (e.g. "*watching movies*", "*watching netflix*", and "*streaming movies*") or context (e.g. "*gaming rig*", which is the term for a device that is designed for playing demanding games, is arguably related to "*computing power*"). Nevertheless, it is true that some clusters are not strongly related to product usage contexts, e.g. "*selling point*" and "*star rating*". Also, there are several inaccuracies, such as "*streaming video*" is clustered together with "*word processing*" and "*video editing*" (which are related to editing jobs), instead of with "*watching movies*". The possible ways to improve this result are optimizing the parameters used in word embedding and creating an additional filtering step to remove the irrelevant bigrams.

At the sentence level, the clustering provides an insight about usage contexts that might appear to be similar. For example, both "*gaming rig*" and "*playing games*" are related to games. However, they are clustered into different groups.

In the cluster that contains "*gaming rig*", the second most frequent word is "*computing power*". It may be seen from the following sentence examples that "*gaming rig*" implies power, which is aligned with the usage contexts that require computing power.

- this was the **gaming rig** , because i wanted power
- in conclusion this laptop is the best **gaming rig** ive ever owned , i needed something for college , and to satisfy my inner gamer
- actually , it's overkill but i'd rather him have too much horsepower than not enough , when it comes to **computing power**
- as far as raw **computing power** goes , again , this is not my desktop , but it is great for an ultrabook form factor

On the other hand, the sentences in the "*playing games*" cluster imply the usage context of playing casual games. The examples of the sentences that contain "*playing games*" are shown below.

- however , its great for browing the web , word processing , and even **playing games** from the windows store (angry birds , wheel of fortune , etc
- very nice , basic , laptop for **playing games** and / or browsing the internet

Interestingly, the cluster that contains "*playing games*" also includes "*demanding games*". A possible explanation is that there are sentences that contain "*demanding games*" which state that the laptop is, in fact, not capable of performing that type of games. As the result, that usage context is clustered with "*playing games*". The examples of sentences that contain "*demanding games*" are shown below.

- again , gaming is possible on the device , but it definitely won't play as good as a device with a dedicated graphics card , so expect performance drawbacks for the more **demanding games**
- they may overclock during gaming but on the more **demanding games** like " dying light " some are reporting a cpu bottleneck with these dual cores

As mentioned by Green *et al*. (2004), the identified usage contexts benefit designers in focusing on gathering customer needs from customers with usage contexts that are considered important and prioritizing the features that support those usage contexts. For example, if most customers are identified to use a laptop for watching movies only, then screen size and resolution might need to be prioritized.

Finally, the result of this paper may be used to assess the discrepancy between manufacturer's (or designer's) intended usage context and customer's actual usage context. In Figure 4, the customer emphasizes that the laptop is for basic usage contexts. On the other hand, the manufacturer's product description states that the product is designed for heavy use, i.e. "The Latitude has always been Dell's flagship model for Notebooks. They were originally **designed for heavy use in the corporate environment**."

The discrepancy may hint several possibilities to be investigated further with respect to the customer whose review is shown in Figure 4, i.e. (1) the customer never needs to perform heavy tasks, (2) the customer had tried to use the laptop for heavy tasks, but he/she was not satisfied, (3) the customer is an outlier among all customers that use the laptop, i.e. most of the customers use the laptop for heavy tasks, or (4) most of the laptops that customers usually use for playing games and web browsing have basic capability that is unable to perform heavy tasks. These possibilities lead to the future work in Section 6.

https://doi.org/10.1017/dsi.2019.257 Published online by Cambridge University Press

6 CONCLUSION AND FUTURE WORK

The paper proposes a methodology that consists of three stages, i.e. Preprocessing, Word Embedding, and Usage Context Clustering to automatically identify and group use cases. In the case study, two clustering methods are compared, i.e. x-means and spherical k-means. The better result is obtained from x-means clustering.

For future work, aspect term sentiment analysis may be added for analyzing customers' sentiment with respect to each usage context. The goal of aspect term sentiment analysis is identifying the sentiment polarity of a target entity which appears in a sentence (Xue and Li, 2018). This enables the creation of a sentiment distribution for each usage context among all products in the market. The distributions would allow designers to locate a product's position compared to its competitors with respect to a particular usage context. Furthermore, designers may also identify potential markets in which most of the current products have obtained negative sentiment from the customers.

REFERENCES

Banerjee, A., Dhillon, I. S., Ghosh, J. and Sra, S. (2005), "Clustering on the Unit Hypersphere Using Von Mises-Fisher Distributions", *Journal of Machine Learning Research*, Vol. 6 pp. 1345–1382.

Belk, R. (1975), "Situational Variables and Consumer Behavior". *Journal of Consumer Research*, Vol. 2 No. 3, pp. 157–164.

Decker, R. and Trusov, M. (2010), "Estimating Aggregate Consumer Preferences from Online Product Reviews", *International Journal of Research in Marketing*, Vol. 27 No. 4, pp. 293–307.

Green, M. G., Palani, R. P. K. and Wood, K. L. (2004), "Product Usage Context: Improving Customer Needs Gathering and Design Target Setting", *ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Salt Lake City, UT, USA.

Green, M. G., Tan, J., Linsey, J. S., Seepersad, C. C. and Wood, K. L. (2005), "Effects of Product Usage Context on Consumer Product Preferences", *ASME 2005 Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Long Beach, CA, USA.

Green, M. G., Linsey, J. S., Seepersad, C. C. and Wood, K. L. (2006), "Frontier Design: A Product Usage Context Method", *ASME 2006 Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Philadelphia, PA, USA.

He, L., Chen, W., Hoyle, C. and Yannou, B. (2012), "Choice Modeling for Usage Context-Based Design", *Journal of Mechanical Design*, Vol. 134 No. 3, pp. 031007. https://doi.org/10.1115/1.4005860

LaFleur, R. S. (1992), "Principal Engineering Design Questions". *Research in Engineering Design*, Vol. 4 No. 2, pp. 89–100. https://doi.org/10.1007/BF01580147

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013a). "Efficient Estimation of Word Representations in Vector Space". *CoRR*, Vol. abs/1301.3781. http://arxiv.org/abs/1301.3781

Mikolov, T., Sutskever, I., Kai, C., Corrado, G. and Dean, J. (2013b), "Distributed Representations of Words and Phrases and Their Compositionality", *26th International Conference on Neural Information Processing Systems - Volume 2*, Curran Associates Inc., USA, pp. 3111–3119.

Novikov, A. (2018), annoviko/pyclustering: pyclustering 0.8.2 release. [online]. Available at: (https://github.com/annoviko/pyclustering) (November 21, 2018). https://doi.org/10.5281/zenodo.1491324.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011), "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830.

Pelleg, D. and Moore, A. W. (2000), "X-means: Extending K-means with Efficient Estimation of the Number of Clusters", *Seventeenth International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 727–734.

Řehůřek, R. and Sojka, P. (2011), "Software Framework for Topic Modelling with Large Corpora", *LREC 2010 Workshop on New Challenges for NLP Frameworks*, ELRA, Valletta, Malta, pp. 45–50.

Rong, X. (2014), "word2vec Parameter Learning Explained". *CoRR*, Vol. abs/1411.2738. http://arxiv.org/abs/1411.2738

Ulrich, K. T. and Eppinger, S. D. (2004), *Product Design and Development*, McGraw-Hill/Irwin, Boston, MA, USA.

Xue, W. and Li, T. (2018), "*Aspect Based Sentiment Analysis with Gated Convolutional Networks", 56th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Melbourne, Australia, pp. 2514–2523.

Zhou, F., Jiao, R. J. and Linsey, J. S. (2015), "Latent Customer Needs Elicitation by Use Case Analogical Reasoning from Sentiment Analysis of Online Product Reviews", *Journal of Mechanical Design*, Vol. 137, No. 7, pp. 071401. https://doi.org/10.1115/1.4030159

https://doi.org/10.1017/dsi.2019.257 Published online by Cambridge University Press