Available online at www.alphanumericjournal.com

alphanumeric journal

The Journal of Operations Research, Statistics, Econometrics and Management Information Systems

Volume 3, Issue 1, 2015

2015.03.01.MIS.01

TEXT MINING AS A SUPPORTING PROCESS FOR VoC CLARIFICATION*

Aysun KAPUCUGİL-İKİZ†

Güzin ÖZDAĞOĞLU‡

Dokuz Eylul University, Faculty of Business, Kaynaklar Kampusu, Buca, Izmir

Received: 18 March 2015
Accepted: 16 May 2015

Abstract

In product development, the foremost issue is to identify "what" the customers' expectations would be from the product. As a promising approach to the product development, Quality Function Deployment also gives crucial importance to the collection and analysis of Voice of the Customer (VoC) to deduce true customer needs. Data sources of VoC include surveys, interviews, focus groups, gemba visits as well as customer reviews which can be collected through call centers, internet homepages, blogs, and microblogs in social networks. Customers' verbatim or reviews obtained from these resources require more detailed extraction to define them as the positive restatement of problems, opportunities or image issues independent of the product or the solution. Basically, this clarification process is a content analysis in which the developers usually seek to extract and classify the spoken-unspoken customer needs from VoC. This labor-intensive manual approach brings subjectivity to the analysis and can take so much time in the case of having condensed and large-volume text data. During the past decade, the field of text mining has enabled to solve these kinds of problems efficiently by unlocking hidden information and developing new knowledge; exploring new horizons; and improving the research process and quality. This paper utilizes a particular algorithm of text clustering, a recently popular field of interest in text mining, to analyze VoC and shows how text mining can also support the clarification process for better extraction of "true" customer needs. Practical implications are presented through analysis of online customer reviews for a product.

Keywords: Voice of the Customer (VoC), Text Mining, Text Clustering, Quality Function Deployment (QFD)

Jel Code: C88, C380, C550, M110, M150, M310

* Initial findings of this paper were also presented in the 20th International Symposium on Quality Function Deployment (ISQFD2014, Istanbul).

† aysun.kapucugil@deu.edu.tr (Corresponding author)

‡ guzin.kavrukkoca@deu.edu.tr

MÜŞTERİ SESİNİN AYRIŞTIRILMASINI DESTEKLEYEN BIR SÜREÇ OLARAK METİN MADENCİLİĞİ

Özet

Ürün geliştirmede en başta gelen konu, müşterilerin üründen beklentilerinin ne olacağını belirlemektir. Ürün geliştirme için gelecek vaadeden bir yaklaşım olarak, Kalite Fonksiyon Göçerimi de, gerçek müşteri ihtiyaçlarını ortaya çıkarmak için Müşteri Sesinin toplanmasına ve analizine oldukça önem vermektedir. Müşteri Sesinin veri kaynaklarını anketler, mülakatlar, odak grupları, gemba ziyaretlerinin yanı sıra çağrı merkezlerinden, internet sayfalarından, web günlüklerinden (blog) ve sosyal ağlardaki mikro web günlüklerinden toplanabilen müşteri yorumları oluşturmaktadır. Bu kaynaklardan elde edilen müşteri ifadeleri veya yorumlarının, ürün ya da çözümden bağımsız problem, fırsat veya imaja yönelik konular bazında yeniden olumlu ifadeler şeklinde tanımlamak için daha detaylı ayrıştırılması gerekmektedir. Temel olarak, bu ayrıştırma süreci, geliştiricilerin genellikle müşteri sesinden dile getirilen ve getirilmeyen müşteri ihtiyaçlarını çıkarmaya ve sınıflandırmaya çalıştıkları bir içerik analizidir. Bu emek-yoğun manuel yaklaşım, analize öznellik getirmekte ve yoğun ve büyük hacimde metin verilerin varlığı durumunda çok fazla zaman alabilmektedir. Son on yılda, metin madenciliği alanı gizli bilgileri açığa çıkararak ve yeni bilgi geliştirerek, yeni ufuklar keşfederek, araştırma sürecini ve kalitesini iyileştirerek bu tür problemlerin etkin bir şekilde çözümüne olanak sağlamaktadır. Bu çalışma, müşteri sesini analiz etmek için, metin madenciliğinin son yıllarda popüler ilgi alanı haline gelen metin sınıflandırmaya yönelik özel bir algoritma kullanmakta ve "gerçek" müşteri ihtiyaçlarını daha doğru bir şekilde belirlemek için metin madenciliğinin ayrıştırma sürecini nasıl destekleyebileceğini göstermektedir. Uygulama açısından etkileri, bir ürüne ilişkin online müşteri yorumlarının analiziyle sunulmaktadır.

Anahtar Kelimeler : Müşteri Sesi, Metin Madenciliği, Metin Sınıflandırma, Kalite Fonksiyon Göçerimi (KFG).
Jel Kodu : C88, C380, C550, M110, M150, M310

1. INTRODUCTION

In product development, the foremost issue is to identify "what" the customers' expectations would be from the product. As a promising approach to the product development, Quality Function Deployment (QFD) also gives crucial importance to the collection and analysis of Voice of the Customer (VoC) to deduce true customer needs.

In the first steps of QFD methodology, a development team tries to segment customers and communicate with them to gather their verbatim through gemba visits, surveys, interviews, focus groups and other tools. In many situations the team can only reach a small group of these customer segments because of time and place limitations. However, today's technology provides many opportunities to customers to share their thoughts and experiences, i.e. customer reviews, about products through consumer web blogs, social networks and even review sections in product web sites. A large volume of customer verbatim can be collected within these platforms in free-text or structured forms. These customer reviews are not only as valuable as the data obtained by interviews and observations but also enable developers to apply advanced data analysis methods to mine hidden patterns in customer wants and needs.

As customers' verbatim or reviews may not directly refer to their needs, they require more detailed extraction to define them as the positive restatement of problems, opportunities or image issues independent of the product or the solution. Basically, this clarification process is a content analysis in which developers usually seek to extract and classify the spoken and unspoken customer needs from VoC based on their judgments in the current QFD practices. However, this labor-intensive manual approach brings subjectivity to the analysis and can take so much time in the case of having condensed and large-volume text data. Besides, it can create a big burden for the developer since the results may include inconsistencies and are prone to failures.

During the past decade, the field of text mining, an interdisciplinary field that draws on information retrieval, data mining, machine learning, statistics, and computational linguistics, has enabled to solve such problems efficiently by unlocking hidden information, developing new knowledge, exploring new horizons, and improving the research process and quality. In this context, text mining refers to the process of deriving high-quality information from text.

Text contains a huge amount of information of any imaginable type and has a major direction and tremendous opportunity for making an inference from customers for better decision making and developing processes. Compared with the kind of data stored in databases, text is unstructured, amorphous, and contains information at many different levels (Witten, 2004). Handling the high volume of free text manually is only feasible by sampling. With the help of algorithms for text mining such as clustering and key term extraction, free-form customer reviews can be processed efficiently and filtered to essential phrases and frequent patterns of content (Gamon et al., 2005).

There are few studies on text mining for handling online customer reviews or verbatim in the context of QFD. Zhao et al. (2005) proposed a visual data mining framework for fast identification of actionable knowledge. By inspiring from the House of Quality (HoQ) which is one of the main elements of QFD, they provided an Opportunity Map which allows the user to focus on classes (i.e. customer requirements) and the values of a particular attribute (i.e. technical or product specification). The relationships between classes and attributes were expressed in various forms, e.g., rules, distributions of data with regard to certain attributes and classes, etc. Zhao et al. (2005) mainly mined actionable rules (opportunities for solving problems) using a class association rule miner. Coussement and Van den Poel (2008) developed a DSS for churn prediction integrating free-formatted, textual information from customer emails with information derived from the marketing database. The emails were preprocessed as a freeform text data to obtain term-document matrix, and the matrix was reduced using semantic and syntactic text mining operations (i.e. latent semantic indexing) to decrease the number of variables for the further analysis. The reduced data set was integrated into the marketing database to apply logistic regression for churn analysis. Therefore the importance of a well considered email handling strategy was highlighted and unstructured call center emails were converted into a more structured form suitable for better identifying customers most prone to switch. Zhang et al. (2010) proposed an Apriori-based data mining approach to extract knowledge from historical data for aiding the designers on the HoQ analysis. The approach they used mainly focused on mining potential useful association rules (including positive and negative rules) that reflect the relationships according to three objectives: support, confidence, and interestingness. However, these studies give no special attention to understand "true" customer needs.

Park and Lee (2011) presented a framework for extracting customer opinions from websites and transforming them into product specification data. They collected customer opinions from an online customer center and then transformed into customer needs using text-mining (i.e. using keyword frequencies about customer complaints and comments). Customers were then segmented into several groups by using k-means clustering algorithm. The relations among customer needs were visualized by co-word analysis and product specifications to meet those needs analyzed by decision tree. Lastly, a final target product specification for new products were determined and a target market was identified based on customer profile data. The study suggested this framework to incorporate customer opinions efficiently with new product development processes, and defined customer needs as product-dependent functional requirements or directly products themselves. However, any statement about the product (i.e. functional or technical requirement) is not a customer need. Functional requirements tell what the product must be or do whereas customer needs are just benefits that tell why things are important to the customers (QFDI, 2013). Customer needs generally lie behind any verbatim or reviews obtained from them. By counting the keyword frequencies only, the true customer needs may not be

extracted from the raw customer complaints, reviews or comments. Their methodology has lack of a component for the customer verbatim clarification process.

Moving from these facts, this paper uses text clustering as a recently popular field of interest in text mining to analyze VoC and shows how this technique can also support the clarification process to extract "true" customer needs from online customer reviews for a product. The paper aims at adding a new approach to the current techniques of QFD methodology and supporting the use of this approach in practice.

The organization of the rest of this paper is as follows. Among other text analytics tasks, Section 2 describes the use of text mining and gives brief information about text clustering. This section gives the details about text clustering process with a special distance-based partitioning algorithm. Section 3 and Section 4 present a case study and its findings on exemplifying text clustering analysis in order to clarify the VoC obtained from online customer reviews and determine the true needs for a particular product. Section 5 presents concluding remarks and suggests future work for the similar research studies.

2. Text Mining

Feldman and Dagan (1995) can be considered as the first study that describes text mining in the context of knowledge discovery and defines it as "knowledge discovery from text" indicating a machine supported analysis of text from information retrieval, information extraction as well as natural language processing integrated with data mining, machine learning and statistics. Text mining concept is handled in different research areas with different perspectives, i.e., information extraction, text data mining, knowledge discovery process (Hotto et al., 2005).

Text mining process simply starts with a collection of documents, then these documents are retrieved and preprocessed by checking format and character sets through a data mining platform including text processing components. When the text data is ready for analysis, a particular method is set or a model is developed to run related algorithms to extract information or to discover a pattern (Gupta and Lehal, 2009).

Traditional text mining applications mostly focus on information access where significant links are searched for finding the right information with the right users at the right time with less emphasis on processing or transformation of text information (Kroeze et al., 2004). .However, the recent text mining applications can be regarded as going beyond information access to further help users analyze information and facilitate decision making.

There are also many applications of text mining where the primary goal is to analyze and discover any interesting patterns in unstructured text data. Such unstructured data can be analyzed at different levels of representation schemes. The simplest representations are bag-of-words and string of words. Text data can also be represented at the level of named entities including people, locations, and organizations to obtain more significant patterns than string of words. Improving text mining algorithms and discovering new patterns also rely on the developments in natural language processing, e.g., information extraction in order to process semantic representations (Mooney and Bunescu, 2005; Aggarwal and Zhai, 2012).

Text mining algorithms have been developed to meet the needs for processing text data collected through many platforms such as social network, web, and other information-centric applications. Research in text mining has an increasing trend with the help of today's advanced hardware and software opportunities. For instance, text mining techniques can be used on large data sets retrieved from the customer reviews to make customer groups or to analyze word intensities to extract some information about their feelings, even customer needs and other categories of items such as problems, solutions and design specifications.

Typical text mining tasks include text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and learning relations between named entities (Han et al., 2012). Main topics under the title "text mining" are:

- Text classification and clustering,

- Information retrieval,
- Information extraction,
- Opinion mining and summarization.

In order to facilitate the clarification of VoC, this paper specifically demonstrates text clustering on the data set retrieved from customer reviews consisting of high volume of free text.

2.1. Text Clustering

Clustering text data has recently become a popular field of interest in data mining. Applications in this area can be categorized as customer segmentation, classification, collaborative filtering, visualization, document organization, and indexing, and so forth. Each of these applications may have different representations of text data (e.g., bag of words, strings) and requires different model designs with different algorithms. Among text clustering algorithms, there is a wide variety of different types such as agglomerative (hierarchical) clustering algorithms, partitioning algorithms, and standard parametric modeling based methods. These clustering algorithms have different advantages and disadvantages when compared in terms of effectiveness and efficiency (Zhao and Karypis, 2004). If hierarchical techniques are used, then nested sequence of partitions are obtained, with a single cluster at the top and unique clusters of individual points at the bottom. Each node in the intermediate level can be considered as merging at least two clusters from the next lower level or splitting a cluster from the next higher level. The result of a hierarchical clustering algorithm can be graphically presented with a dendogram. In contrast to hierarchical techniques, partitional clustering techniques create partitions of data points at just one level. If K is the desired number of clusters, then partitional approaches typically find all k clusters at once around centroid points. (Steinbach et al., 2000)

The purpose of the study is finding clusters for user reviews collected around particular words then finding needs of different customer groups rather than finding hierarchy of the words or customers. This necessity and the quadratic time complexity are the underlying cause of selecting one of the partitional clustering techniques in the study. The next section explains the details of partitional clustering techniques used in text clustering.

2.2. Partitional Clustering Tecniques

Partitioning algorithms are especially used in order to obtain clusters of objects. For text clustering, the two most widely used partitioning algorithms are k-medoids and k-means clustering algorithms based on distances between documents (Aggarwal and Zhai, 2012).

In k-medoid clustering algorithms, a set of points is used as medoids around which the clusters are built. The aim of the algorithm is to determine an optimal set of representative documents from the original document collection around which the clusters are constructed. Finally, each document is assigned to its closest representative from the document collection. K-medoid algorithms are relatively slow because of the number of iterations they compute and they may not produce satisfactory results when there are not many common words in large document set (Aggarwal and Zhai, 2012).

The k-means clustering algorithm also uses a set of k representatives around which the clusters are constructed. However, these representatives are not necessarily obtained from the original data and are refined somewhat differently than a k-medoids approach. The simplest form of the k-means approach is to start off with a set of k key items from the original document collection, and assign documents to these items on the basis of closest similarity. The centroid of the assigned points to each seed is used to replace the seed in the last iteration. In the next phase, the new key item is defined, if it is a better central point for this cluster. This approach is processed until the required convergence level is achieved. One of the advantages of the k-means method over the k-medoids method is that it requires an extremely small number of iterations in order to converge (Cutting, et al., 1992). Because of this advantage, k-means clustering method is preferred in the application of this paper.

K-means algorithm is applied through the following steps (Kwale, 2013):

1. Apply text preprocessing (tokenization, filtering stopwords, transform cases, stemming, n-grams, converting text to numeric data, etc.)
2. Choose the number of clusters, k
3. Randomly generate k clusters and determine the cluster centers (centroids), where a cluster's centroid is the mean of all points in the cluster.
4. Repeat the following until no object moves (i.e. no object changes its cluster)

4.1. Determine the distance of each object to all centroids. (Cosine similarity is used in the study to calculate distances)

4.2. Assign each point to the nearest centroid.

4.3. Re-compute the new cluster centroids.

K-means clustering is an unsupervised learning technique that can be used to understand the underlying structure in a dataset. When used with text data, k-means clustering can provide a great way to organize bag of words used by customers to describe their visits. Once it is clear what the customers are trying to do, then it is not only possible to transform these experiences to match the needs, but also to adjust reporting/dashboards to monitor the various customer groups (Harvard TagTeam, 2013).

The most important parameter of k-means algorithm is "the number of clusters (k)" which can directly affect the performance of the results. The performance of this algorithm is generally measured by two indexes, i.e., average silhouette and Davies Bouldin indexes. The higher the silhouette index, or the lower the Davies Bouldin index, the better the partitioning among the clusters (Turi and Ray, 2000; Chandrasekhar, et al., 2011).

3. Case Study: Text Clustering on Customer Reviews

This paper analyzes the user reviews of a particular model of "XYZ diaper bag" obtained from a popular website publishing user reviews for several types of products.

On the web-site, XYZ diaper bag is described as

"….transports all the necessary features for an efficient parenting supply source, including changing pad, key clip and stabilizing metal feet. The mode includes plenty of pockets inside and out for organizing, separating and keeping things clean and in their place. Depending on preference and situation, the mode allows for three carrying options including two tote straps that snap together for easy carrying and stow away when not in use, a removable padded shoulder strap, or stroller attachments for a no-slip grip to your stroller handle."

Pictures given in Figure 1 demonstrate the functionality of this bag. The web site that publishes reviews on the product has an effective screen design to support customers. In addition to user reviews in free-text form, positive and negative review categories and ratings are also presented for each product (Figure 2).

For analyzing the customer verbatim on XYZ diaper bag, 232 user reviews with its titles and ratings were downloaded in a sheet by the help of particular software. Then text data mining process was designed on Rapid Studio® 6.0x as follows.

Text Preprocessing

Text clustering process starts with preprocessing of the free-form text data so that advance analysis techniques can be used for further inferences. After the data file was imported to Rapid Studio®, common text-preprocessing techniques were used to clean the data for further analysis (Figure 3). In this step, the following operations were applied:

a. **Tokenization** is the process of decomposing a complete text into words, phrases, symbols, called tokens. The list of tokens is then used as input data for further processing. Tokenization is implemented based on non-letters and linguistic sentences within the partial model presented as Figure 3.
b. **Filtering stop words** is applied to clean data from the characters or useless words, e.g., punctuators and prepositions.
c. **Filtering tokens** is applied to select the character range to consider, i.e., words 2 to 25 characters.
d. **Transforming cases** is used to convert all words into lowercase.

e. **Stemming** is the process of cleaning the suffixes to combine word coming from the same root, e.g., go, goes, going. Porter's stemming method is used within the partial model presented as Figure 3.

f. **Creating n-gram** provides to analyze more than one word repetition. "Generate n-grams" operator in Figure 3 generates word repetitions up to three words.

g. **Creating term vector** based on TF-IDF method (see the equations 1 to 3). This parameter is defined within the "process documents from data" operator in the upper level of the model (Figure 4 and Figure 7).

h. **Pruning** is applied to reduce the time complexity by eliminating words/word groups below 10%. This parameter is defined within the "process documents from data" operator in the upper level of the model (Figure 4 and Figure 7).

For an effective clustering process, the word frequencies should be normalized in terms of their relative frequencies calculated according to the occurrences in the document and over all documents in the data set. In general, a common representation used for text processing is the vector-space based TF-IDF representation (Salton, 1983). In the TF-IDF representation, the term frequency (TF) for each word is normalized by the inverse document frequency (IDF). The IDF normalization reduces the weight of terms which occur more frequently in the collection. This reduces the importance of common terms in the collection, ensuring that the matching of documents be more influenced by those of more discriminative words which have relatively low frequencies in the collection (Jane and Dubes, 1998).

TF-IDF is the product of two statistics: term frequency and inverse document frequency. There are various ways for determining the exact values of both statistics. In the case of the term frequency tf(t,d), the number of times that term t occurs in document d is divided by the number of occurrences of the most frequent word (w) within the same document (Equation 1).

Figure 1. XYZ Diaper Bag
Source: Buzzilions (2014), "XYZ diaper bag" Review Statistics

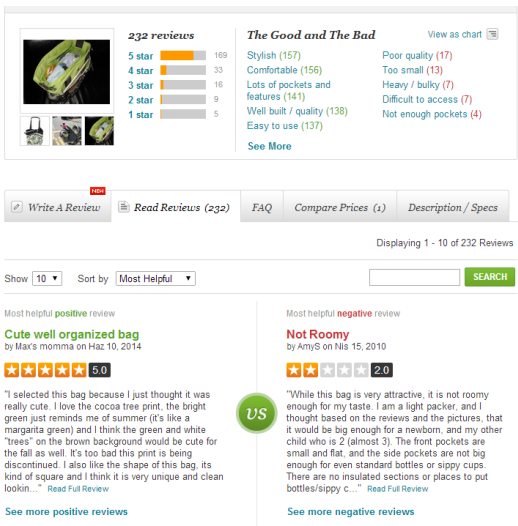


Figure 2. Customer Reviews' Sample from the Web-page
Source: Buzzilions (2014), "XYZ diaper bag" Review Statistics

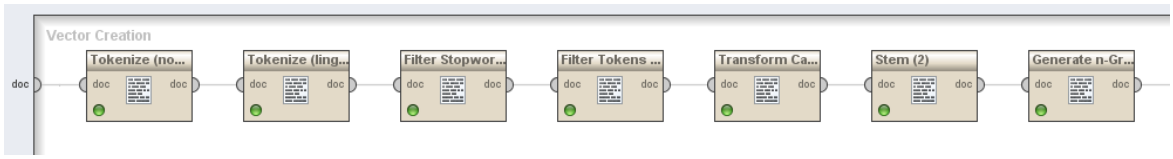


Figure 3. Text Preprocessing

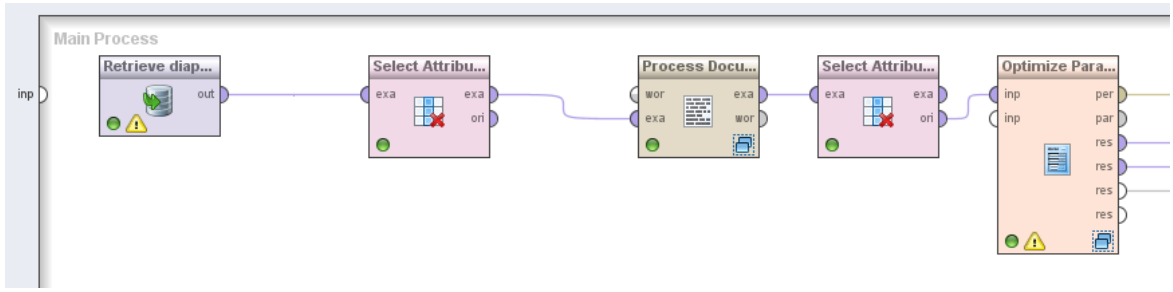


Figure 4. Text Clustering with Parameter Optimization

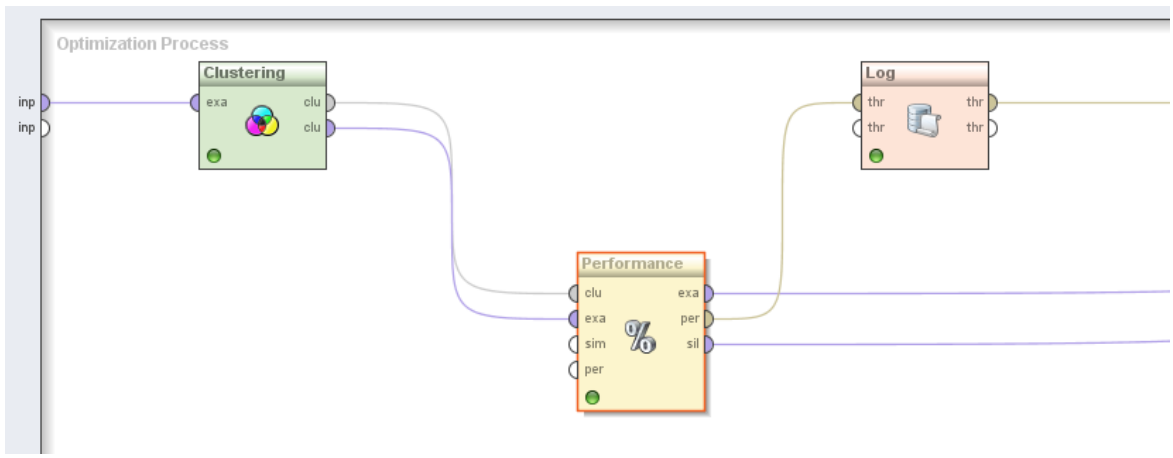


Figure 5. Nested Process in Parameter Optimization Block

$$tf(t,d) = \frac{f(t,d)}{max\{f(w,d):w\in d\}} \quad (1)$$

IDF is a measurement of how much information the word provides, that is, whether the term is common or rare across all documents. It is the logarithmically scaled fraction of the documents that contain the word, obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient (Equation 2).

$$idf(t,D) = log\frac{N}{|\{d\in D:t\in d\}|} \quad (2)$$

where N is the total number of documents in the document collection, $|\{d \in D: t \in d\}|$ is the number of documents where the term t appears.

Thus, *TF-IDF* can be calculated as follows (Equation 3):

$$tf - idf(t,d,D) = tf(t,d) \times idf(t,D) \quad (3)$$

Text Clustering

In this paper *k*-means clustering algorithm, a well-known technique among distance-based partitioning

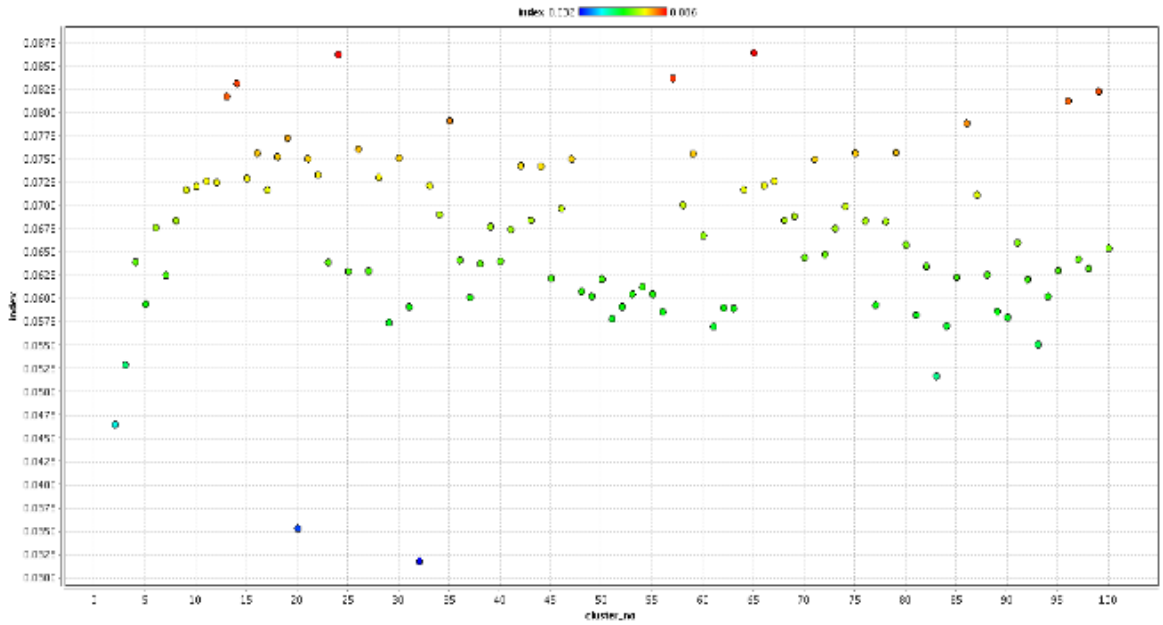


Figure 6. Performance Index Values with Respect to the Number of Clusters

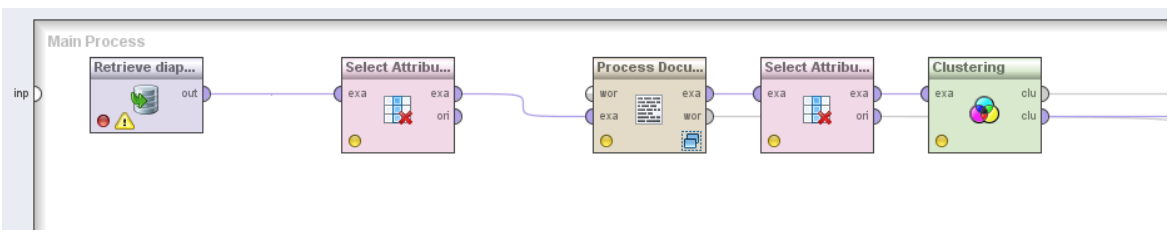


Figure 7. K-means Text Clustering Based on 25 Clusters

algorithms is preferred for analyzing the data set due to the advantages explained in Section 2.2.

The most important parameter of *k*-means algorithm is the number of clusters that is the value of *k.* In order to find the best value for the number of customers, parameter optimization was applied with respect to "silhouette index", i.e., an appropriate performance indicator for *k*-means clustering, and the best value was obtained when 25 clusters were applied. The corresponding screens for parameter optimization and comparison of different *k* values are given in Figure 4, Figure 5 and Figure 6.

The number of clusters (k) can be measured by two indexes, i.e., average silhouette or Davies Bouldin indexes, as mentioned in Section 2.2. As both can produce similar results, the silhouette index (denoted as Index in Figure 6) is used to determine k in this study. A higher Index value within its range indicates the number of clusters which can partition the data set in clearer way. Figure 6 shows an example scatter plot produced in Rapid Studio® for the relationship between Index and k. Index value is found between 0.032 and 0.086 based on the nature of the data used in analysis. The highest of value of this index suggests choosing k as 24. However, it should be pointed out that the minor changes on the selection of pruning percentage even may affect the value of silhouette index. For this study, several pruning percentages are observed in the neighborhood of 10%. In these trials, Index values resulted in similar k values between 20 and 30. Authors chose k as the middle point of this range and therefore, K-Means clustering algorithm is

applied through the "clustering" operator within the model shown in Figure 7 considering 25 clusters that are determined based on Euclidean Distances. The corresponding parameter tab is also presented in Figure 8 to clarify the options selected in Rapid Studio®.

Text clustering was applied on the data set to visualize customer clusters in accordance with items which potentially indicate feeling, need, or product specifications. The output table can easily be interpreted and integrated to the existing tools in QFD.

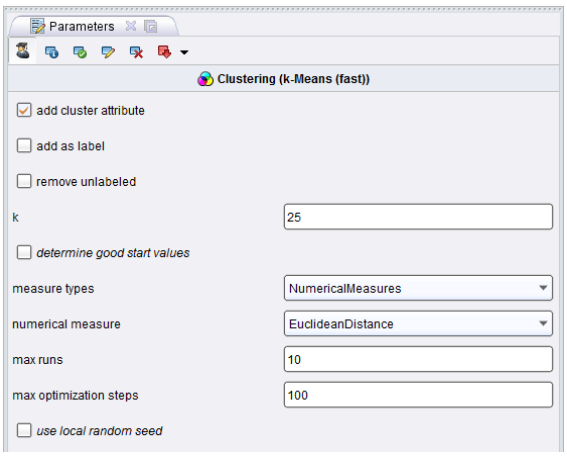


Figure 8. Parameter Tab for K-means Clustering

4. Findings

Customer ratings published on the web site indicate that most of the customers are generally satisfied from the product (Figure 9).

Figure 9. Distribution of Customer Ratings

Source: Buzzilions (2014), "XYZ diaper bag" Review Statistics

On the related web site customer reviews are published with tagged words/terms which also provide some aggregated findings from all review contents (Figure 10).

A word cloud is also developed using Wordle® (2013) to highlight the most frequent words in the source text and represented in Figure 11.

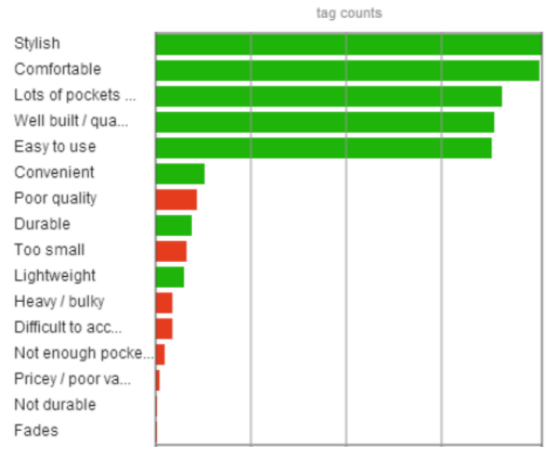


Figure 10. Good and Bad Tag Counts

Source: Buzzilions (2014), "XYZ diaper bag" Review Statistics

In addition to these statistics, text preprocessing performed in the study provides a word list or bag of words (see Appendix 1) from which frequencies of

Figure 11. Word Cloud of the Data Set

words can be utilized for observing what kind of expressions were highlighted over the frequencies, e.g., impressions, problems, product specifications, or even needs. In the word list, each word or word group constitutes an attribute. "Total occurrences" of a word is the attribute's frequency value for the document set. "Document occurrences" refer to the number of documents that include the corresponding word or the attribute.

Text preprocessing also generates a term vector, where the rows indicate customers or documents, and

the columns indicate the words/terms, i.e. attributes. In the intersection of the rows and the columns, the cells correspond to TF-IDF values that describe the intensity of the term/word in the corresponding document. Appendix 2 shows partial representation of the term vector. The term vector is then used as a numerical data set to perform text clustering on the customer with respect to the particular words they have used in the reviews. K-means clustering is applied on the term vector to see the possible partitions among the customers based on distances between the term frequencies. For this purpose, k-means clustering produces centroid tables (Appendix 3) to show the partitioning structure on the term vector.

Clusters obtained from the text mining phase are analyzed based on the highlighted words repeated in each cluster. These are clarified items which are only simplified, single-issue expressions of what the customer says, behaves, or documents. As mentioned in Introduction Section, customer needs are just benefits that tell why things are important to the customers and generally lie behind any verbatim or reviews obtained from them. The true customer needs cannot be extracted from the raw customer complaints, reviews or comments only by counting their frequencies. An analyst should go beyond the stated ones in order to extract the true needs in turn create differentiated products. This extraction can be done by analysts by exploring the value for the customer, which is vital step of any QFD work and especially VoC clarification process. Authors try to extract some meaningful categorizations of expressions based on the clusters. In this regard, customer needs, product specifications, and impressions are extracted and rewritten with respect to concepts provided by QFD methodology (Table 1, Table 2, and Table 3). These results also show the common phrases among the clusters. For instance, some of the customers highlight the features associated with fashion whereas others indicate its functionality (Table 1). Beside the needs, customers also comment on the specifications of the product, i.e. pockets to organize materials, carrying options, and size (Table 2). Finally, Table 3 indicates that customers have positive feelings about the product.

Table 1. Extracted Customer Needs

Clusters	Customer Needs	Number of Reviews
0, 1, 2, 5, 7, 9, 12, 15, 16, 19, 21, 24	I can fit many things	
0, 1, 7, 8, 10, 17, 21, 23	Options to carry the bag	
6, 8, 12, 17, 21,	Easy to clean	
3, 11, 13, 14, 16, 17	I want to keep things organized	
3, 4, 14, 19, 24	I want a stylish bag	
2, 11, 20,	Big enough to carry	
8, 19	Durable	
8, 16	Easily hold a full size bottle	

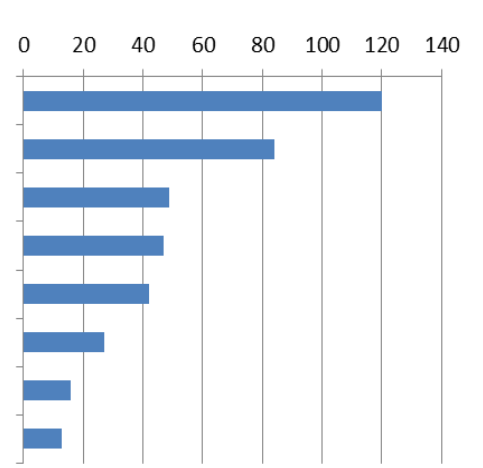


Table 2. Highlighted Product Specifications

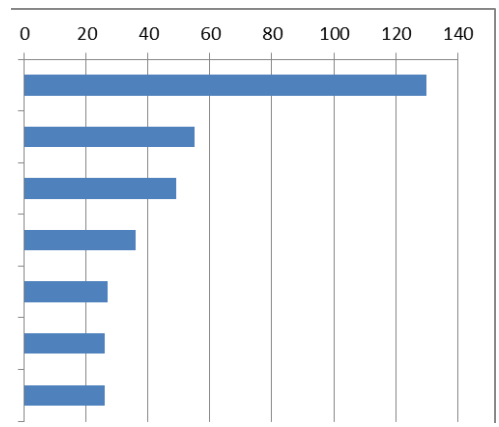
Clusters	Product Specifications	Number of Reviews
0, 3, 6, 8, 9, 11, 13, 16, 17, 18, 19, 20, 21, 22	Multiple purpose pockets	
0, 7, 10, 17, 21	Hangings for the stroller	
5, 6, 14, 18, 21	Changing pad	
10, 21, 23	Padded shoulder strap	
3, 6, 16, 20	Size	
3, 7, 24	Separate compartments	
9, 18	Adjustable handles (straps)	

Table 1. Impressions Declared in the Reviews

Clusters	Impressions	Number of Reviews
6, 16, 20	perfect	20
0, 9, 14, 17, 19, 24	great	67
0, 3, 7, 10, 11, 20, 23	love	60
5, 8, 12	cute	28
21, 22	nice	20

When compared with the product reviews published on the website, it can be observed that extracted needs and highlighted product specifications are compatible with the tag statistics, but the web site statistics do not indicate the true needs so that product designers can use them in the improvement process. This inadequacy can be removed by adopting QFD methodology that provides designers such an approach to understand customer needs for better product improvements. However, analysis of a large-scale text data through the current QFD techniques will increase the time complexity of applications. In this context, this paper proposes that integrating text mining with QFD methodology over customer reviews will create a powerful and agile decision support to identify and categorize clarified items from customers' verbatim in order to discover hidden patterns among these items for better extraction of

5. Conclusion and Future Remarks

QFD gives crucial importance to the collection and analysis of VoC to deduce true customer needs and suggests many analytical tools to extract the needs from customer verbatim. Customer verbatim is traditionally collected from various sources such as surveys, interviews, focus groups, gemba visits. As networking platforms arise in the past decade, customers have the opportunity to write or declare their reviews through call centers, internet homepages, blogs, and microblogs in social networks. These reviews provide more verbatim than the traditional data collection methods and reach larger data sizes that need advance analysis approaches like text mining and web mining algorithms.

Text mining refers to the process of deriving high-quality information from text. This paper discussed text mining to identify and categorize clarified items from customers' verbatim in order to discover hidden patterns among these items for better extraction of customer needs. A case study was conducted on the customers' reviews of a diaper bag in order to show how text preprocessing and clustering can help for VoC clarification. K-means text clustering, a commonly used clustering algorithm based on distance-based partitioning, was applied on the

customer reviews in free text form, and the most frequent group of words and customer clusters were obtained from over 200 reviews in a few minutes. These statistics were used to define clarified items which therefore led to extracting customer needs, highlighted product specifications, and customer's overall impressions about the product. The findings also indicated that starting VoC analysis process through text mining provided objective approach for the further phases of QFD. Besides this advantage, analysts should consider that a deviation might appear if many fake reviews are published to increase or decrease the popularity of the product.

Consequently, this paper discussed the use of text mining as a supporting tool within QFD. Text mining provided many advantages in analyzing VoC over customer reviews. The study may be improved by adding different text mining tools such as other clustering approaches as well as association or classification algorithms to further analyze the details of customer reviews and need extraction. Experimental analysis may also be conducted with different parameter sets and new learning schemes may be added to improve the extraction skills in text analytics. This study emphasizes the idea that text mining techniques really support and accelerate to analyze VoC using QFD philosophy.

QFD adds a value with its knowledge base and tailored approach to retrieve meaningful extractions from customer reviews through text mining. In order to extract automatically the customer needs through text mining, it is necessary to form a dynamic corpus specific to the product under consideration. As future work, such a corpus may be formed in a long run project and besides, learning algorithms may be improved using some insights from QFD.

References

Aggarwal, C.C., Zhai, C.X.(Eds), (2012). Mining Text Data, Springer, New York, e-ISBN 978-1-4614-3223-4.

Buzzilions, (2014). "XYZ Diaper Bag" Review Statistics, http://www.buzzillions.com/diaper-bag-reviews, Access Date: August 18th, 2014.

Chandrasekhar, T., Thangavel, K., Elayaraja, E., (2011). Performance analysis of enhanced clustering algorithm for gene expression data, International Journal of Computer Science Issues, 8(6-3), 253-257.

Coussement, K., Van den Poel, D., (2008). Integrating the voice of customers through call center emails into a decision support system for churn prediction, Information & Management, 45: 164–174.

Cutting, D.R., Pedersen, J.O., Karger, D.R., Tukey, J.W., (1992). Scatter/Gather: A cluster-based approach to browsing large document collections. SIGIR '92: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, ACM Press, 318-329.

Feldman, R., Dagan, I., (1995). KDT-Knowledge Discovery in Texts. Proceedings of the First International Conference on Knowledge Discovery, 112-117.

Gamon, M., Aue, A., Oliver S.C., Ringger, E.,(2005). Pulse: Mining customer opinions from free text, Advances in Intelligent Data Analysis VI, 6th International Symposium on Intelligent Data Analysis, IDA, Proceedings, Madrid, Spain, September 8-10, Springer (Lecture Notes in Computer Science), 121-132.

Gupta, V., Lehal, G.S., (2009). A survey of text mining techniques and algorithms, Journal of Emerging Technologies in Web Intelligence, 1(1):60-76.

Han, J., Kamber, M., Pei, J., (2012). Data Mining, Concepts and Techniques, Morgan Kaufmann Publishers, Waltham, MA, USA.

Harvard TagTeam, (2013). http://tagteam.harvard.edu/hub_feeds/1981/feed_items/274117. Access: 01.08.2014

Hotto, A., Nirnberger, A., Paaβ, G., Augustin, S., (2005). A Brief Survey of Text Mining, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.210.158&rep=rep1&type=pdf, Access date: 05.05.2014, pp.4-5.

Jane, A.K., Dubes, R.C., (1998). Algorithms for Clustering Data, PrenticeHall, Englewood Cliffs, NJ.

Kroeze, J.H., Matthee, M.C., Bothma, T.J.D., (2004). Differentiating between data-mining and text-mining terminology. South African Journal of Information Management, 6(4): 93-101.

Kwale, F.M., (2013). A critical review of k means text clustering algorithms. International Journal of Advanced Research in Computer Science, 4(9): 1-9.

Mooney, R.J., Bunescu, R., (2005). Mining knowledge from text using information extraction, SIGKDD Explorations, 7(1): 3-10.

Park, Y., Lee, S., (2011). How to design and utilize online customer center to support new product concept generation, Expert Systems with Applications, 38: 10638–10647.

QFDI (2013). Quality Function Deployment Institute, QFD Black Belt® Workshop, Course Workbook, Chapter 13: Analyze Customer Voice, 9 -13 September 2013, SantaFe NM, USA.

Salton, G., (1983). An Introduction to Modern Information Retrieval, Mc-Graw Hill, 1983.

Steinbach, M., Karypis, G., Kumar, V.,(2000). A comparison of document clustering techniques. KDD Workshop on Text Mining. http://glaros.dtc.umn.edu/gkhome/node/157, Access date: March 4th, 2014.

Turi, R.H., Ray, S., (2000). Determination of the Number of Clusters in Colour Image Segmentation, SCSSE Monash University, Clayton Vic Australia.

Witten, I.H., (2004). Adaptive text mining: inferring structure from sequences, Journal of Discrete Algorithms, 2:137–159.

Wordle, (2013). http://www.wordle.net/. Access date: 05/01/2015.

Zhao, K., Liu, B., Tirpak,T.M., Xiao, W., (2005). Opportunity map: A visualization framework for fast identification of actionable knowledge, CIKM'05, October 31-November 5, 2005, Bremen, Germany, http://www.cs.uic.edu/~kzhao/Papers/05_CIKM05_kaidi.pdf, Access date: January 31st, 2014.

Zhao, Y., Karypis, G., (2004). Empirical and theoretical comparisons of selected criterion functions for document clustering, Machine Learning, 55(3):311–331.

Zhang, Z., Cheng, H.,Chu, X., (2010). Aided analysis for quality function deployment with an Apriori-based data mining approach, International Journal of Computer Integrated Manufacturing, 23(7): 673–686.

Appendix 1. Word List (bag of words)

Word (Attribute)	Total Occurrences	Document Occurrences	Word (Attribute)	Total Occurences	Document Occurrences	Word (Attribute)	Total Occurrences	Document Occurrences
attach	40	38	go	30	25	room	49	43
babi	86	66	got	43	36	shoulder	45	41
bag	573	213	great	80	56	shoulder strap	28	26
big	55	45	hold	51	39	size	41	34
bottl	44	31	keep	39	36	space	29	27
bought	46	37	look	68	59	strap	91	67
carri	72	55	lot	65	56	stroller	93	74
chang	71	59	love	225	148	stuff	43	37
chang pad	58	53	love bag	49	47	stylish	25	24
clean	35	34	month	36	33	thing	79	59
come	33	28	nice	38	29	us	103	81
compart	29	26	old	37	29	want	34	26
cute	41	39	organ	38	37	wipe	34	32
diaper	212	127	pad	80	62			
diaper bag	167	112	perfect	46	40			
durabl	26	24	plenti	36	31			
easi	59	53	pocket	170	110			
fit	55	44	put	31	23			
get	52	47	recommend	31	31			

Appendix 2. Partial Representation of the Term Vector

Review Id	Cluster	attach	babi	bag	big	bottl	bought	carri	chang	chang pad	clean	come	compart	cute	diaper	diaper bag	durabl	easi	fit	get	go	got	great	hold	keep	look	lot	love	love_bag	month	nice	old	organ	pad
1	cluster_4	0.161	0.000	0.038	0.000	0.000	0.654	0.128	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.065	0.202	0.000	0.000	0.142	0.000	0.000	0.507	0.000	0.000	0.122	0.127	0.040	0.000	0.000	0.000	0.000	0.000	0.000
2	cluster_21	0.101	0.070	0.067	0.091	0.449	0.000	0.161	0.229	0.247	0.214	0.118	0.000	0.298	0.067	0.041	0.127	0.000	0.093	0.089	0.124	0.000	0.079	0.000	0.000	0.076	0.079	0.025	0.000	0.000	0.232	0.000	0.205	0.368
3	cluster_0	0.105	0.219	0.035	0.190	0.233	0.000	0.250	0.079	0.086	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.086	0.193	0.185	0.000	0.000	0.247	0.000	0.000	0.000	0.247	0.078	0.000	0.226	0.000	0.121	0.106	0.076
4	cluster_3	0.000	0.247	0.034	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.429	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.366	0.000	0.000	0.176	0.000	0.000	0.000	0.000	0.360	0.000
5	cluster_7	0.276	0.192	0.065	0.000	0.000	0.560	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.184	0.222	0.000	0.450	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	cluster_24	0.000	0.000	0.057	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.201	0.243	0.000	0.000	0.000	0.000	0.000	0.000	0.311	0.000	0.000	0.000	0.000	0.300	0.266	0.000	0.000	0.000	0.000	0.000
7	cluster_13	0.000	0.000	0.041	0.393	0.483	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.144	0.087	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.223	0.328	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
8	cluster_5	0.000	0.000	0.042	0.403	0.000	0.000	0.000	0.337	0.000	0.000	0.000	0.000	0.439	0.000	0.000	0.000	0.000	0.000	0.393	0.000	0.000	0.000	0.000	0.000	0.337	0.000	0.000	0.000	0.000	0.512	0.000	0.000	0.000
9	cluster_7	0.333	0.000	0.047	0.000	0.000	0.338	0.000	0.000	0.000	0.353	0.000	0.000	0.000	0.333	0.402	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.083	0.000	0.359	0.000	0.000	0.000	0.000
10	cluster_6	0.000	0.000	0.036	0.000	0.425	0.388	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.254	0.154	0.000	0.000	0.000	0.000	0.000	0.000	0.300	0.000	0.000	0.000	0.000	0.190	0.337	0.000	0.000	0.000	0.000	0.000
11	cluster_10	0.000	0.000	0.063	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12	cluster_23	0.000	0.000	0.026	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.567	0.000	0.000	0.000	0.000	0.432	0.137	0.000	0.000	0.000	0.000	0.000
13	cluster_12	0.000	0.000	0.016	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.324	0.109	0.132	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
14	cluster_11	0.000	0.000	0.030	0.000	0.000	0.000	0.250	0.000	0.000	0.333	0.000	0.000	0.000	0.105	0.126	0.000	0.256	0.000	0.277	0.000	0.000	0.000	0.000	0.647	0.000	0.000	0.156	0.000	0.000	0.000	0.000	0.319	0.000
15	cluster_24	0.000	0.416	0.019	0.000	0.000	0.000	0.000	0.151	0.163	0.000	0.000	0.000	0.000	0.067	0.080	0.250	0.000	0.000	0.176	0.000	0.000	0.000	0.000	0.197	0.000	0.151	0.000	0.050	0.000	0.000	0.000	0.000	0.146
16	cluster_13	0.000	0.000	0.028	0.000	0.330	0.000	0.000	0.224	0.242	0.000	0.000	0.000	0.000	0.296	0.119	0.000	0.000	0.272	0.000	0.000	0.000	0.000	0.000	0.305	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.216
17	cluster_23	0.000	0.000	0.016	0.000	0.000	0.000	0.527	0.000	0.000	0.000	0.000	0.400	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.408	0.000	0.000	0.000	0.000	0.000	0.000	0.164	0.292	0.000	0.000	0.000	0.336	0.000
18	cluster_2	0.000	0.296	0.020	0.386	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.142	0.000	0.000	0.000	0.000	0.000	0.000	0.438	0.000	0.000	0.000	0.000	0.000	0.106	0.000	0.000	0.000	0.000	0.000	0.000
19	cluster_5	0.000	0.000	0.021	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.439	0.000	0.000	0.000	0.363	0.000	0.393	0.000	0.000	0.000	0.000	0.000	0.000	0.350	0.221	0.000	0.000	0.000	0.000	0.000	0.000
20	cluster_1	0.000	0.000	0.040	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.277	0.094	0.113	0.000	0.000	0.000	0.000	0.000	0.000	0.277	0.000	0.000	0.000	0.000	0.070	0.000	0.000	0.000	0.324	0.000	0.000
21	cluster_19	0.000	0.151	0.031	0.000	0.000	0.000	0.000	0.000	0.000	0.231	0.000	0.000	0.000	0.000	0.000	0.272	0.177	0.000	0.000	0.000	0.000	0.512	0.000	0.224	0.329	0.171	0.000	0.000	0.000	0.000	0.000	0.220	0.000
22	cluster_0	0.000	0.000	0.019	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.324	0.406	0.000	0.000	0.000	0.000	0.000	0.444	0.000	0.474	0.000	0.000
23	cluster_8	0.000	0.000	0.014	0.000	0.000	0.000	0.233	0.000	0.000	0.000	0.342	0.000	0.289	0.000	0.000	0.367	0.000	0.000	0.000	0.000	0.302	0.000	0.000	0.000	0.222	0.000	0.073	0.258	0.631	0.000	0.000	0.000	0.000
24	cluster_18	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25	cluster_3	0.000	0.000	0.025	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.640	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.131	0.467	0.000	0.000	0.000	0.000	0.000
26	cluster_6	0.000	0.000	0.014	0.090	0.330	0.000	0.000	0.225	0.161	0.000	0.000	0.359	0.000	0.033	0.000	0.000	0.000	0.182	0.087	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.100	0.217
27	cluster_2	0.000	0.000	0.036	0.343	0.000	0.000	0.301	0.000	0.000	0.000	0.221	0.000	0.000	0.252	0.228	0.000	0.154	0.174	0.167	0.000	0.000	0.000	0.000	0.000	0.143	0.000	0.094	0.000	0.000	0.217	0.434	0.192	0.000
28	cluster_21	0.000	0.000	0.070	0.000	0.412	0.000	0.294	0.280	0.302	0.393	0.000	0.000	0.000	0.000	0.000	0.000	0.302	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.425	0.000	0.270
29	cluster_17	0.000	0.000	0.017	0.000	0.400	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.239	0.145	0.000	0.293	0.000	0.000	0.000	0.000	0.354	0.370	0.000	0.000	0.000	0.089	0.000	0.000	0.000	0.000	0.365	0.000
30	cluster_21	0.000	0.156	0.042	0.000	0.000	0.000	0.000	0.170	0.183	0.000	0.262	0.000	0.000	0.149	0.181	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.170	0.000	0.056	0.000	0.000	0.516	0.000	0.000	0.327
31	cluster_12	0.000	0.000	0.071	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.334	0.404	0.000	0.410	0.000	0.443	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32	cluster_12	0.000	0.000	0.042	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.149	0.180	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.339	0.000	0.111	0.000	0.000	0.000	0.000	0.000	0.000
33	cluster_2	0.199	0.692	0.038	0.361	0.000	0.000	0.158	0.000	0.000	0.000	0.000	0.000	0.000	0.133	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.049	0.000	0.215	0.000	0.229	0.000	0.000
34	cluster_17	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.472	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.454	0.000	0.000	0.000	0.000	0.000	0.000	0.000
35	cluster_10	0.000	0.000	0.028	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.690	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.294	0.521	0.000	0.000	0.000	0.000	0.000
36	cluster_14	0.000	0.000	0.019	0.000	0.000	0.000	0.000	0.299	0.322	0.000	0.462	0.478	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.098	0.000	0.000	0.000	0.000	0.000	0.288
37	cluster_3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.341	0.000	0.388	0.317	0.000	0.000	0.000	0.000	0.000	0.284	0.000	0.000	0.000	0.000	0.000	0.331	0.243	0.253	0.000	0.000	0.000	0.000	0.000	0.000

Appendix 3. Centroid Table

words	clstr0	clstr1	clstr2	clstr3	clstr4	clstr5	clstr6	clstr7	clstr8	clstr9	clstr10	clstr11	clstr12	clstr13	clstr14	clstr15	clstr16	clstr17	clstr18	clstr19	clstr20	clstr21	clstr22	clstr23	clstr24
attach	0.051	0.067	0.066	0.000	0.050	0.048	0.088	0.162	0.000	0.087	0.064	0.000	0.000	0.025	0.061	0.000	0.079	0.039	0.000	0.000	0.000	0.033	0.036	0.052	0.044
babi	0.070	0.031	0.373	0.174	0.022	0.062	0.049	0.065	0.045	0.023	0.045	0.085	0.039	0.014	0.109	0.074	0.000	0.031	0.000	0.063	0.125	0.069	0.000	0.066	0.166
bag	0.033	0.031	0.035	0.017	0.042	0.040	0.037	0.057	0.025	0.030	0.037	0.031	0.035	0.032	0.024	0.054	0.030	0.024	0.318	0.041	0.036	0.036	0.045	0.037	0.034
big	0.013	0.000	0.248	0.000	0.000	0.106	0.071	0.021	0.117	0.145	0.031	0.042	0.000	0.057	0.041	0.051	0.121	0.000	0.000	0.000	0.100	0.073	0.000	0.020	0.000
bottl	0.039	0.000	0.000	0.000	0.000	0.000	0.120	0.000	0.160	0.033	0.000	0.000	0.000	0.100	0.021	0.047	0.418	0.057	0.000	0.066	0.000	0.115	0.112	0.021	0.037
bought	0.000	0.000	0.033	0.000	0.582	0.000	0.145	0.162	0.000	0.129	0.026	0.065	0.000	0.037	0.000	0.000	0.000	0.000	0.000	0.084	0.031	0.023	0.066	0.000	0.024
carri	0.049	0.054	0.209	0.056	0.030	0.066	0.000	0.052	0.026	0.056	0.000	0.134	0.059	0.045	0.000	0.067	0.000	0.067	0.074	0.047	0.033	0.089	0.029	0.209	0.000
chang	0.005	0.000	0.000	0.000	0.047	0.176	0.251	0.017	0.074	0.055	0.019	0.023	0.036	0.090	0.251	0.000	0.000	0.000	0.165	0.029	0.000	0.136	0.000	0.029	0.042
chang_pad	0.006	0.000	0.000	0.000	0.000	0.122	0.262	0.019	0.048	0.053	0.021	0.025	0.039	0.040	0.239	0.000	0.000	0.000	0.178	0.000	0.000	0.147	0.000	0.032	0.018
clean	0.000	0.029	0.000	0.049	0.054	0.000	0.000	0.106	0.000	0.012	0.060	0.086	0.153	0.021	0.000	0.171	0.038	0.045	0.000	0.084	0.032	0.123	0.043	0.000	0.075
come	0.033	0.000	0.020	0.041	0.023	0.111	0.050	0.000	0.038	0.078	0.107	0.036	0.000	0.023	0.147	0.000	0.000	0.000	0.067	0.000	0.087	0.038	0.042	0.046	0.000
compart	0.022	0.000	0.034	0.303	0.040	0.000	0.040	0.051	0.054	0.028	0.054	0.000	0.079	0.000	0.154	0.069	0.000	0.054	0.000	0.025	0.000	0.011	0.000	0.049	0.000
cute	0.027	0.083	0.063	0.045	0.074	0.415	0.066	0.029	0.178	0.009	0.034	0.000	0.175	0.000	0.023	0.000	0.000	0.000	0.000	0.021	0.030	0.021	0.040	0.022	0.067
diaper	0.024	0.052	0.125	0.026	0.108	0.078	0.087	0.201	0.090	0.089	0.031	0.041	0.115	0.111	0.045	0.194	0.065	0.082	0.077	0.120	0.084	0.055	0.112	0.109	0.083
diaper_bag	0.019	0.063	0.105	0.031	0.107	0.081	0.057	0.225	0.043	0.103	0.028	0.049	0.110	0.093	0.054	0.234	0.039	0.060	0.093	0.110	0.063	0.058	0.106	0.116	0.087
durabl	0.000	0.034	0.026	0.000	0.022	0.034	0.000	0.048	0.396	0.014	0.081	0.000	0.000	0.025	0.000	0.000	0.000	0.000	0.000	0.152	0.000	0.026	0.051	0.000	0.028
easi	0.022	0.022	0.085	0.095	0.042	0.068	0.000	0.099	0.000	0.024	0.093	0.145	0.158	0.030	0.015	0.000	0.029	0.395	0.040	0.064	0.025	0.107	0.000	0.012	0.058
fit	0.287	0.156	0.042	0.039	0.055	0.049	0.053	0.000	0.017	0.046	0.000	0.000	0.000	0.081	0.000	0.251	0.000	0.041	0.045	0.064	0.028	0.044	0.033	0.045	0.093
get	0.056	0.093	0.071	0.072	0.033	0.212	0.010	0.040	0.058	0.000	0.000	0.031	0.066	0.017	0.061	0.000	0.187	0.037	0.000	0.019	0.037	0.037	0.125	0.073	0.096
go	0.000	0.028	0.054	0.000	0.125	0.000	0.000	0.000	0.023	0.109	0.032	0.077	0.000	0.000	0.000	0.314	0.000	0.000	0.000	0.000	0.000	0.081	0.000	0.071	0.000
got	0.037	0.069	0.121	0.000	0.020	0.000	0.000	0.000	0.034	0.000	0.000	0.000	0.065	0.028	0.098	0.043	0.000	0.000	0.050	0.000	0.123	0.023	0.142	0.376	0.059
great	0.099	0.018	0.010	0.000	0.082	0.000	0.081	0.000	0.014	0.214	0.055	0.078	0.070	0.049	0.438	0.000	0.000	0.139	0.000	0.159	0.000	0.049	0.000	0.013	0.132
hold	0.061	0.040	0.000	0.000	0.019	0.073	0.000	0.044	0.292	0.063	0.000	0.100	0.000	0.025	0.018	0.165	0.184	0.051	0.000	0.120	0.251	0.012	0.000	0.037	0.022
keep	0.000	0.092	0.000	0.306	0.000	0.000	0.000	0.024	0.020	0.000	0.000	0.472	0.000	0.093	0.000	0.150	0.151	0.093	0.000	0.072	0.000	0.044	0.000	0.019	0.039
look	0.072	0.034	0.044	0.127	0.091	0.105	0.026	0.132	0.025	0.057	0.019	0.024	0.108	0.062	0.049	0.067	0.027	0.000	0.038	0.178	0.032	0.041	0.142	0.068	0.045
lot	0.108	0.043	0.032	0.138	0.014	0.145	0.049	0.000	0.064	0.054	0.020	0.036	0.032	0.083	0.064	0.095	0.062	0.269	0.045	0.139	0.000	0.023	0.032	0.145	0.000
love	0.111	0.062	0.057	0.104	0.092	0.043	0.087	0.135	0.105	0.046	0.133	0.079	0.063	0.040	0.052	0.025	0.081	0.033	0.035	0.046	0.103	0.078	0.055	0.113	0.093
love_bag	0.211	0.063	0.049	0.104	0.074	0.000	0.075	0.034	0.104	0.011	0.216	0.120	0.030	0.000	0.023	0.050	0.032	0.000	0.000	0.042	0.027	0.048	0.000	0.139	0.069
month	0.137	0.025	0.055	0.000	0.021	0.029	0.033	0.142	0.132	0.066	0.000	0.035	0.000	0.064	0.000	0.105	0.039	0.000	0.000	0.000	0.057	0.000	0.000	0.079	0.025
nice	0.000	0.077	0.020	0.000	0.000	0.127	0.000	0.026	0.022	0.028	0.000	0.000	0.000	0.000	0.000	0.073	0.091	0.000	0.000	0.079	0.048	0.173	0.611	0.000	0.111
old	0.106	0.073	0.216	0.049	0.070	0.000	0.000	0.067	0.070	0.074	0.000	0.000	0.000	0.031	0.000	0.000	0.000	0.000	0.050	0.000	0.000	0.077	0.000	0.000	0.027
organ	0.045	0.000	0.030	0.147	0.042	0.000	0.079	0.023	0.019	0.020	0.000	0.341	0.000	0.039	0.024	0.043	0.000	0.193	0.000	0.031	0.000	0.050	0.000	0.026	0.000
pad	0.005	0.017	0.009	0.063	0.000	0.109	0.298	0.017	0.057	0.053	0.019	0.022	0.035	0.036	0.214	0.000	0.000	0.000	0.159	0.035	0.000	0.276	0.000	0.028	0.016
perfect	0.000	0.027	0.000	0.136	0.042	0.000	0.215	0.074	0.039	0.037	0.000	0.000	0.098	0.100	0.026	0.086	0.303	0.000	0.000	0.020	0.475	0.012	0.000	0.018	0.032
plenti	0.000	0.000	0.035	0.000	0.000	0.045	0.012	0.043	0.000	0.061	0.000	0.000	0.100	0.022	0.000	0.051	0.000	0.000	0.339	0.035	0.000	0.082	0.000	0.025	0.341
pocket	0.065	0.064	0.021	0.068	0.048	0.059	0.128	0.009	0.131	0.099	0.025	0.112	0.043	0.244	0.050	0.064	0.100	0.141	0.062	0.094	0.146	0.138	0.159	0.080	0.023
put	0.071	0.029	0.085	0.000	0.025	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.085	0.588	0.034	0.000	0.000	0.054	0.000	0.000	0.000	0.000	0.000	0.000	0.071
recommend	0.031	0.095	0.033	0.000	0.135	0.053	0.040	0.069	0.000	0.030	0.000	0.000	0.045	0.025	0.000	0.000	0.223	0.043	0.069	0.035	0.000	0.095	0.000	0.023	0.068
room	0.050	0.051	0.067	0.110	0.096	0.000	0.020	0.138	0.054	0.024	0.032	0.059	0.000	0.088	0.054	0.052	0.000	0.109	0.000	0.077	0.000	0.027	0.038	0.080	0.202
shoulder	0.026	0.133	0.000	0.049	0.000	0.000	0.011	0.019	0.018	0.000	0.119	0.039	0.071	0.035	0.000	0.105	0.000	0.000	0.089	0.046	0.000	0.242	0.000	0.149	0.000
shoulder_strap	0.033	0.083	0.000	0.000	0.000	0.000	0.013	0.000	0.023	0.000	0.150	0.000	0.000	0.024	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.295	0.000	0.067	0.000
size	0.045	0.000	0.059	0.178	0.019	0.000	0.218	0.021	0.020	0.074	0.000	0.000	0.081	0.000	0.000	0.000	0.269	0.000	0.000	0.101	0.236	0.045	0.095	0.000	0.000
space	0.000	0.000	0.000	0.000	0.060	0.000	0.000	0.047	0.000	0.011	0.000	0.000	0.517	0.032	0.000	0.000	0.000	0.000	0.109	0.000	0.036	0.116	0.043	0.045	0.000
strap	0.035	0.063	0.035	0.084	0.000	0.019	0.008	0.065	0.042	0.095	0.534	0.049	0.000	0.025	0.000	0.000	0.125	0.000	0.098	0.041	0.000	0.229	0.000	0.147	0.000
stroller	0.062	0.086	0.071	0.070	0.032	0.093	0.049	0.251	0.025	0.116	0.134	0.041	0.025	0.028	0.037	0.000	0.115	0.127	0.068	0.030	0.042	0.079	0.064	0.033	0.028
stuff	0.000	0.000	0.136	0.043	0.000	0.116	0.011	0.000	0.000	0.029	0.035	0.000	0.025	0.100	0.000	0.157	0.080	0.000	0.044	0.511	0.000	0.009	0.000	0.082	0.058
stylish	0.000	0.050	0.047	0.172	0.022	0.000	0.000	0.000	0.000	0.060	0.000	0.096	0.000	0.000	0.166	0.000	0.000	0.000	0.063	0.000	0.000	0.000	0.000	0.000	0.329
thing	0.384	0.000	0.000	0.081	0.046	0.026	0.023	0.112	0.050	0.018	0.023	0.227	0.083	0.141	0.000	0.113	0.074	0.000	0.037	0.045	0.078	0.017	0.049	0.000	0.111
us	0.058	0.053	0.067	0.056	0.050	0.036	0.078	0.116	0.021	0.064	0.231	0.019	0.019	0.026	0.049	0.137	0.077	0.260	0.062	0.062	0.217	0.093	0.166	0.089	0.013
want	0.000	0.629	0.047	0.000	0.022	0.000	0.000	0.000	0.047	0.012	0.000	0.000	0.067	0.033	0.037	0.056	0.000	0.000	0.000	0.000	0.064	0.016	0.050	0.086	0.040
wipe	0.000	0.000	0.066	0.000	0.045	0.000	0.178	0.030	0.136	0.000	0.033	0.000	0.027	0.054	0.000	0.116	0.000	0.364	0.000	0.053	0.000	0.047	0.084	0.000	0.000