Customer Feedback Analysis Applying the RFC Model, Hybrid Sentiment Extraction and Contrast Dictionary

Myint Zaw

A Thesis Submitted in Fulfillment of the Requirement for the Degree of

Doctor of Philosophy in Computer Engineering

Prince of Songkla University

2019

Thesis Title	Customer Feedback Analysis Applying the RFC Model, Hybrid Sentiment Extraction and Contrast Dictionary
Author	Mr. Myint Zaw
Major Program	Computer Engineering

Major Advisor	**Examining Committee:**
……………...	……………............................... Chairperson
(Asst. Prof. Dr. Pichaya Tandayya)	(Assoc.Prof. Dr. Ponrudee Netisopakul)
	……………................................. Committee
	(Asst. Prof. Dr. Pichaya Tandayya)
	……………................................. Committee
	(Asst. Prof. Dr. Panyayot Chaikan)
	……………................................. Committee
	(Dr. Somchai Limsiroratana)
	……………................................. Committee
	(Dr. Anant Choksuriwong)

The Graduate School, Prince of Songkla University, has approved this thesis as fulfillment of the requirements for the Degree of Doctor of Philosophy in Computer Engineering.

………………………………………………...

(Prof. Dr. Damrongsak Faroongsarng)
Dean of Graduate School

This is to certify that the work here submitted is the result of the candidate’s own investigations. Due acknowledgement has been made of any assistance received.

……... Signature
(Asst. Prof. Dr. Pichaya Tandayya)
Major Advisor

……... Signature
(Mr. Myint Zaw)
Candidate

I hereby certify that this work has not been accepted in substance for any degree, and is not being currently submitted in candidature for any degree.

…………....................................... Signature

(Mr. Myint Zaw)

Candidate

Thesis Title	Customer Feedback Analysis Applying the RFC Model, Hybrid Sentiment Extraction and Contrast Dictionary
Author	Mr. Myint Zaw
Major Program	Computer Engineering
Academic Year	2019

ABSTRACT

In recent years, the marketers have been moving to sale and trade their products in social markets due to the growth of Internet technologies. This allows the consumers to generate, share, criticize, and feedback on their interested products. The customer feedbacks come from many sources and the data become massive. The so-called Social Market Big Data (SMBD) involves many types of customers' expressions which are valuable to extract their opinions and satisfactions to support the marketers to make better decisions. However, the characteristics of SMBD are represented in unstructured formats which usually are only human readable. Manual information extraction is time consuming and labor-intensive. In this study, a new algorithm multi-level sentiment analysis to extract the customer opinions from feedbacks, called contrast dictionary is proposed. It can perform better than two well-known algorithms, the SentiStrength and Word-count, especially on negative feedbacks. Furthermore, a new hybrid approach to extract sentiments which improves the performance of sentiment extraction algorithms, called the aspect-based sentiment information extraction is also proposed.

Moreover, the marketers require more comprehensive information on customer perspectives concerning the products and services comparing with the others. Previously, there had been no studies to grade the products based on the comparison of extracted information from SMBD. In relation to that, the RFM (Recency, Frequency, and Monetary) model is a measurement technique to compare market information, especially in traditional market analytics. This research also proposes a new approach by modifying the RFM model to classify the products from SMBD, called the RFC (Recency, Frequency, and Credit) model. The model focuses on the social market information and product categorization based on the customer satisfactions from customer feedbacks.

The performance of the contrast dictionary has been validated with well-known sentiment information extraction algorithms, SentiStrength and Word-count, tested on Yelp and Amazon review polarity datasets. The proposed algorithm yields the 76.09% accuracy on the Yelp dataset, comparing with the 73.79% accuracy of SentiStrength and the 69.38% accuracy of Word-count. Also, it yields the 72.58% accuracy on the Amazon dataset that is more correct than the SentiStrength and Word-count those yield the 69.68% and 67.35% accuracies respectively. Furthermore, the proposed hybrid sentiment approach can improve the accuracy about 6.12% on the training dataset and 11.67% on the testing dataset. The RFC model produces new knowledges from customer feedbacks on products to be applied on a decision support and recommendation system in marketing management.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude and sincere appreciation to the following persons and organizations, who have helped, supported, and encouraged directly or indirectly towards the completion of this thesis possible.

First and foremost, I am deeply grateful to my advisor, Assistant Professor Dr. Pichaya Tandayya for her invaluable technical supervision, advice, encouragement, and guidance to make the thesis a success. I also thank her for advising and commenting many issues to improve my thesis which include reading, writing, presenting, critical thinking, and so forth.

It is my pleasure to express my deepest thankfulness to all lecturers in Prince of Songkla University who enriched valuable knowledge and guidance during the period of study. I sincerely thank Dr. Somchai Limsiroratana and Asstant Professor Dr. Panyayot Chaikan who taught, pointed, and encouraged me during my study.

I would like to thank my beloved parents, brothers, and sisters who always help me with what I need and who always give me constant encouragements. Also, I am grateful to Bing Liu for the positive and negative words dataset, and Mike Thelwall for the SentiStrength algorithm.

My special thanks to two of my beloved friends in Thai, Mr. Bukhoree Sahoh from Signal Processing Lab and Mr. Maykungth Metha from Personal Health Record Lab, who were always beside and encouraged me when I got any depressions. My study could not succeed without their helps, especially Mr. Bukhoree Sahoh. All my colleagues from Parallel and Distributed Computing Lab such as Mr. Attasuntorn Traisuwan, Mr. Taofik Lamsub, and Mr. Thada Wangthammang, for their invaluable suggestion and helpful contributions. Also, I would like to thank all staff in Department of Computer Engineering, especially Miss Bongkot Prucksapong for dealing with many documents and the conference traveling plan during the period of study.

I would like to acknowledge my scholarship from the Thailand's Education Hub for Southern Region of ASEAN Countries Project Office of the Higher Education Commission. Also, I am grateful for the thesis research funding from PSU Graduate School. The financial supports of both organizations have helped me to come and study in PSU which can fulfill the requirements of my research. Furthermore, I would like to thank the funding from Faculty of Engineering and Department of Computer Engineering for my trip to participate in the international conference.

Finally, my special thanks to all related to my thesis those I could not possibly list. I would like to acknowledge the helps, supports, encouragements, and inspirations received from other nice peoples who directly or indirectly contributed what they could towards the success of this thesis.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS AND SYMBOLS

SMBD	Social Marketing Big Data
SNS	Social Network Services
RFM	Recency, Frequency, and Monetary
RFC	Recency, Frequency, and Credit
NLP	Natural Language Processing
OM	Opinion Mining
DM	Data Mining
ML	Machine Learning
RB	Rule Based
API	Application Programming Interface
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative

CHAPTER 1
INTRODUCTION

This chapter provides an overview of the coverage and structure of this research that consists of four main parts. First is the introduction and motivation as shown in Section 1.1. Then, the objectives of the thesis are represented in Section 1.2 and the scopes are listed in Section 1.3. Finally, the benefits of the study are explained in Section 1.4.

1.1 Problem Statements and Motivations

The need to understand customer satisfactions on products and services is a basic business objective for designing better marketing strategies and improving their products and services effectively [1]. These strategies can support marketing personnel and experts in making better decisions [2]. Traditionally, marketing companies discover the patterns of customer behaviors by using the customers' transactions in databases and data warehouse [3], [4]. These can help identify company's approach strategies based on customer satisfactions. In relation to that, the RFM (Recency, Frequency, and Monetary) model is a well-known methodology for processing and analyzing customer data in order to discover customer behavior patterns [1], [4], [5]. A key aspect of the RFM model is to analyze customer behaviors about how recent a customer has purchased their products (R), how frequent a customer purchases their products (F), and how much a customer spends the total money for their products (M) [1], [4], [5]. The RFM model in terms of market analytics have been widely used, e.g., analyzing churn players for online games [6], analyzing veterinary hospital customer values [5], classifying the bank customers [3], and sales and trade marketing strategies [7]. Although the RFM model is in flavor of utilization in various companies for analyzing patterns of customer behaviors, its applications still focused on direct data and did not cover the customer satisfactions from indirect or external data (e.g., customer feedbacks).

Strictly pointed to external data, social networking services (SNS) such as Facebook, Twitter, and YouTube are very popular amongst the Internet users or customers as they are now used in daily life [8], [9]. This is because advanced technologies such as mobile phones, tablets, and smart devices are available to support the customers to generate, share, and reuse data in anywhere at any time [10], [11]. Normally, millions of people share their experiences on the social media [12]. This can be seen in the case of Twitter, it has more than 326 million active users monthly, and 500 million tweets per day, and every second the users tweet more than 7000 messages [13], [14]. In this situation, the social media data has specific characteristics, namely, high-speed generating, various types (such as text, audio, image, and video), diversified sources of data, and the amount becomes massive, called Social Marketing Big Data (SMBD). SMBD consists of many kinds of user expressions associated with their opinions. For example, they complain about services and products, criticize current phenomena and also show engagements [15] such as likes, comments, and shares through multi-media represented in unstructured data such as free-text [16] in the social data stream [11]. In terms of economic marketing, the customer opinions on SNS play as part of historical data [17], [18]. Therein, the challenges to find a hidden knowledge in SMBD have recently contributed to a broad research area to track and monitor customer opinions [9], [19], [20], [21] in order to support the marketing strategies. Therefore, marketers may use SMBD to understand the customer feedbacks on products and services. From that perspective, SMBD has an important role in the alternative marketing sustainability in understanding their company's market, investigating for their customer satisfactions. Although the SMBD consists of customer opinion data that can enable decision-makers to discover the knowledge, the characteristics of SMBD are difficult to be extracted [21], [22], [23]. Therefore, the aforementioned major problem is a new challenge. The main research question to examine is "How can we analyze SMBD to understand the customer feedbacks for supporting the decision-makers in the field of economic marketing?"

In recent years, there have been increasing interests in tracking and monitoring customer opinions over SMBD that associate with consumer engagements, for instance, to analyze and predict political events [24], [25], [26], and also to analyze the customer complaints in terms of management services [17], as well as to analyze a word-of-mouth concerning movie reviews [18], [27], to analyze customer acceptances [28], to identify the influence customer's attitudes and opinions [20], to identify consumer resonance for purchase intention [29], to analyze popular places and suitable locations for new shops [30], and to analyze stock movement prediction of publicly listed companies [16], [31], [32]. As the mentioned literatures, they used several techniques to extract the customer opinions from SMBD and confirmed that their approaches can support the marketing. For example, S. Tartir and I. Abdul-Nabi [33] proposed an approach to get sentiment information using the bag-of-word approach combining with the semantic understanding from the Arabic language. They proved their extracted information compared with manually extracted information by three Arabic native speakers. Correspondingly, L. I. Tan and et.al [34] introduced a rule-based approach to extract the sentiment polarity (positive or negative) using sentiment analysis from financial-textual news. However, they did not have rules to extract the sentiment for complex sentences and focused only on sentence-level sentiments. Moreover, these are not enough to handle well the complexity of human languages usage in customer feedbacks. The sentences from customers' feedbacks may not be grammatically correct and can also be complex. They often use a lot of contrast words in one feedback [35], [36]. To solve this problem, this research proposes a new sentiment analysis algorithm, called the contrast dictionary that works based on multiple levels: aspect, word, sentence or phrase, contrast, and document. It can handle the complexity of human language more than the well-known algorithms, SentiStrength and Word-count.

Multi-level sentiment information extraction is a technique that can provide the information more accurately than a single level [37], [38]. In the past decade, most of researches proposed the sentiment extraction only on a document level using several techniques [9], [10], [39]. T. Chen and et.al [40] and X. Fang and J. Zhan [41] proposed word-level, sentence-level, and document-level sentiment extraction. They proved that the sentiment information from multiple levels improved the extraction performance. However, they do not consider contrast information. Herein, our algorithm includes all levels of sentiment information to decide for a feedback sentiment.

On the other hand, the marketers, researchers and producers are interested in extracting the aspect-level sentiment information. The aspects of products include cost, quality, and so forth [37], [42]. B. B. Alengadan and S. S. Khan [37] proposed an approach to extract the aspect-level sentiment information extraction from feedbacks. They extracted information based on the five targeted aspects of K5 Kia Motors which are design, performance, price, quality, and service. They used well-known machine learning techniques such as Term Frequency–Inverse Document Frequency (TF/IDF), Support Vector Machine (SVM), and Random Forest. Z. Piao and et.al [42] introduced an aspect based approach extracted from customer feedbacks to rank the product providing useful information for marketers and customers.

Sentiment extraction and aspect-level sentiment extraction from SMBD have been studied by many researchers producing many algorithms and techniques. However, they had not been applied together. Some feedbacks may not contain any desired aspects and only sentiment information are mentioned and vice versa. Moreover, they concern different levels of information extraction. Sentiment extraction works in the document level but aspect-based sentiment works in the word and sentence levels [43]. They also use different techniques. The sentiment extraction can apply dictionary-based and machine learning techniques [9], [44]. However, the previous usage of aspect sentiments applied the machine learning-based techniques and always depended on the aspect features [37], [42], [43]. E. Cambria [38] explained a hybrid approach to extract the sentiment information which provided more effective results on information extraction. However, they did not technically combine the sentiment and aspect sentiment information extraction from SMBD. Therefore, this study also proposes a new hybrid sentiment approach, called the aspect-based sentiment extraction, to fuse the results of both approaches for more accurate and intuitive sentiment information.

The customer opinion is information that can support the marketers to understand their products in SMBD. In practice, they do not need just opinions from feedbacks. According to the literatures, most of previous works focused on discovering the knowledge from SMBD for customer engagements but lacked of co-considered relations among them as of the RFM model capabilities. It is interesting to apply the advantages of the RFM model on data retrieving from SMBD in order to understand the customer feedbacks on products and services. However, the product information extraction usually depends on customers' opinions extracted from the customers' feedbacks [39], [42]. Actually, customers give the product credits on the SMBD by showing their opinions in feedbacks. The credit (C) value of opinions can replace the monetary (M) value in the RFM model. Therefore, we also propose the RFC (Recency, Frequency, and Credit) model to segment the customer opinions on products from feedbacks on SMBD. In the SMBD context, the Recency (R) represents the date and time of the last customer's feedback, the Frequency (F) is the total number of feedbacks, and the Credit (C) shows the sum of all customers' sentiments on a product during a certain period.

The information based on the recency and frequency could be flexible and reusable in forecasting [45], [46], [47], [48]. It can help reveal the dynamics of information of products and services more deeply. On the contrary, S. Du, et.al [45] introduced the recency effectiveness based on customer corporation in social networks to the marketers. They represented corporate social responsibility which also showed that it could be used in the economic marketing. Moreover, C. Kendall [47] proposed the efficiency of information based on time. They suggested that the information based on recency can provide long-term information more efficiently. This means that the information based on time segments can provide reusable and flexible information for the future. In reality, the recency information works as a supporter to help understand the information of current and past situations [4]. Therefore, the study of SMBD analytics based on the RFC is needed to apply in fields of economic marketing in order to provide covered information. The challenge in this field is "How to interpret the customer feedbacks from external data such as the SMBD based on the RFC model to understand product information in the real-world perspective?"

These three variables can help classify products and services in the same way that the RFM grades customer behaviors. Although the proposed RFC model may be a good technique to categorize products, it cannot be directly used in SMBD. It is completely different from traditional marketing data due to the characteristics of SMBD. It represents in unstructured texts which only human can interpret and understand, not in a machine-readable form. To apply the RFC model in SMBD, social marketing, the advanced technologies for processing unstructured data and converting it into a machine-readable form are required.

For that requirement, NLP is the technology to process and analyze unstructured data from the SMBD, and transform it into a machine-readable form [49], [50]. For instance, T. Chen, R. Xu, Y. He, and X. Wang [40] proposed the approach for extracting data to understand human opinions using the natural language processing. The output from the approach showed that it was possible to support the machine to understand the unstructured data effectively. In this point, NLP is the key solution for extracting essential variables which are the R and F from SMBD for being used in the RFC model.

Likewise, the second technique is Opinion Mining. It is the concept of mining the hidden patterns of human language (the output from former NLP) in order to uncover the information from social user feedbacks and complaints [10], [42]. For example, C. Catal and M. Nangir [51] and L. Bing et al. [16] provided the Opinion Mining approach for discovering the hidden information based on the customer feedback sentiments from SMBD. The outputs from these approaches can support the decision-makers to understand the customer feedbacks deeply. In this situation, Opinion Mining is also the potential technique for recognizing the information to apply and support the RFC model.

To analyze that information from the aforementioned technologies, Data Mining (DM) techniques provide differentiation among three variables (R, F, and C) for extracting new knowledge [4], [5], [6], [52]. It is the backbone for the RFC analytics and can also be interpreted in the same way as the RFM model. As an example, D. Birant [4] and M. Mohammadian and I. Makhani [52] proposed an DM approach to extract hidden knowledge from the RFM model and they confirmed that this knowledge can apply in real-world marketing strategies. From that point of view, the DM is one of the main technologies for extracting the pattern or knowledge of customer opinions in order to support the marketing strategies based on the RFC model.

From these perspectives, the advantage of NLP (Information Extraction) is a challenge to apply in terms of computing to pre-process and transform SMBD into a proper structured data for counting to support both Recency and Frequency which are needed by the RFC model. To find out the Credit, Opinion Mining (OM) is the methodology for uncovering the valuable information from SMBD finding the positive value and negative value, to support the last variable of the RFC model. The interpretation of the relationship among those variables (Recency, Frequency, and Credit) helps analyze the influential products of marketing strategies. DM provides the analytical data model for an automated approach to extract the information among them based on the RFC model. Finally, this analytical data model can be reasoned for dividing and grouping the products based on the integration of the knowledge from customer feedbacks and the RFC variables applying rule-mining techniques (one of DM technique).

The integration of key technologies namely, NLP, OM, and DM are the main technologies to support the RFC model for analyzing the customer feedbacks to apply the RFC model in SMBD analytics. This is a new approach for SMBD analytics in terms of social marketing for deeply understanding the customer feedbacks on products and services. In the study, we proposed a new opinion mining algorithm and the social market analytics using the RFC model combining with the Data Science Technologies: NLP, OM, and DM. The RFC model can also be applied in social market analytics and decision support systems. Furthermore, it is possible to apply it in an alarming system to monitor the trends of products.

1.2 Objectives

1.2.1 To propose a new dictionary-based sentiment extraction algorithm in order to handle the complexity of human languages in free texts from SMBD including ungrammatical and complex usages.

1.2.2 To improve the information extraction performance of customer opinions on products by fusing aspect-based sentiment analysis.

1.2.3 To propose the RFC model for analyzing customer satisfactions and categorizing products using NLP, OM and DM.

1.3 Scopes

1.3.1 The input is customer feedback data from Amazon that consists of 142.8 million review data spanning May 1996 - July 2014 [53], [54].

1.3.2 The output is the extracted information from SMBD concerning the customer opinions and types of products based on their perspectives.

1.3.3 The system can process only textual data written in English.

1.3.4 The proposed sentiment extraction algorithm is tested on the two different big labelled-datasets [55].

1.3.5 The RFC model to categorize the products based on the customer satisfactions is tested on the product reviews from the amazon.com which associated with datasets of “Automotive Products.”

1.4 Benefits

1.4.1 The decision-makers get the summarized knowledge by interpreting the sentiment polarity (positive and negative) from the customer opinions of products on SMBD using the RFC model. It can be useful for marketing management, decision support and recommendation systems.

1.4.2 The marketers can use the aspect sentiment extraction to understand their customers’ opinions on products’ aspects.

1.4.3 The tracking, monitoring, notification, and alarming systems on customer satisfactions can use the contrast dictionary algorithm to notify the negative feedbacks and can use the RFC model to provide a notification on information changes in a time period.

1.4.4 The hybrid approach can provide the better accuracy of information extraction to the users.

CHAPTER 2
LITERATURE REVIEWS

This chapter provides the summarization of previous literatures on customer feedbacks and market analysis system using data science techniques. Section 2.1 describes the usage of the RFM model in traditional market analysis. Section 2.2 represents the definition and overview of Natural Language Processing. The sentiment information extraction using Opinion Mining is shown in Section 2.3. Then, the Data Mining techniques especially applying in the RFM model is discussed in Section 2.4. Finally, the summary of the mentioned literatures is represented in Section 2.5.

2.1 The RFM (Recency, Frequency, and Monetary) Model

In traditional market analysis, the RFM model is a famous technique to analyze the customer behaviors introduced by Bult and Wansbeek in 1995 [3], [4]. A key aspect of the RFM model is to analyze customer behaviors about how recently the customers has purchased their products (R), how frequent the customers purchase their products (F), and how much the customer spends for the products (M) [4], [6] from the customer transactions in database or data warehouse. Furthermore, the key strength of the RFM model is collecting and detecting the customer behaviors for a long duration in the traditional market. For instance, the marketing manager can easily know the customers whether they currently are active or passive customers based on the recency (R) value. Moreover, the frequency (F) and monetary (M) values can easily divide the level of customers.

Customer behaviors have been studied by many researchers using the RFM model in traditional market analysis systems. In the history of economics development, the RFM has been thought of as a key factor in economic and marketing. Y. S. Patel, D. Agrawal, and L. S. Josyula [3] proposed a framework to analyze the behaviors of customers. It used RFM analytics for analyzing the bank customer behaviors. Consequently, they showed the percentages of the customer value categorization: high values, medium values, and low values. Similarly, J. T. Wei, S. Y. Lin, Y. Z. Yang, and H. H. Wu [5] proposed a system to analyze the customer values of a veterinary hospital. They also used the RFM analytics combining with the K-means clustering method. The results classified the customers into four categories (spender, average, frequent, and uncertain). Furthermore, E. G. Castro and M. S. G. Tsuzuki [6] proposed a system to analyze the churn online game customers. They used K-NN classification algorithm based on the RFM model. Although the RFM model had been researched by many researchers, the model has lots of advantages and disadvantages based on the literatures [1], [3], [4], [5], [7], [52], [56] as discussed in Table 2.1.

Table 2.1. The advantages and disadvantages of the RFM model in traditional marketing.

No.	Advantages	Disadvantages
1	The RFM model is a high-performance technique for identifying the characteristics of customer behaviors known as the behavior patterns.	The RFM model aims to define the behavior of customers from only internal data but does not cover external data such as SMBD.
2	The RFM model supports the organization for understanding their customers using their pattern, it can help hands on the promotion to the right customer at the right time.	It focuses on internal transactions with a small number of variables but does not consider the attitudes of the customers such as customer emotions or sentiments.
3	The RFM model requires small memory computation. This is because it uses only three main variables but can conclude the pattern of customer behaviors.	Their variables focus on only individual customers but do not include the crowdsourcing perspectives such as the perspectives from groups of the Internet users providing social media reactions.
4	The RFM model requires only specific transaction variables and does not involve more personalized customer preferences.	The RFM model focuses only on one-to-many (only one customer buys many products many times) but ignores the case of many-to-one (many customers buy the same products many times).
5	The RFM model is the long-term technique to deeply understand the customer using the correlation among R, F, and M. It can categorize the brand-loyal customers.	That model does not concern how to cope with the customer feedbacks in a real-time system such as customer complaints or satisfactions about the products on the micro-blog or Internet forum.

From aforementioned perspectives, the limitations of the RFM model generally associate with the variables of well collected and formatted internal data and does not involve with unstructured data (external data or customer feedbacks). The challenges are to apply the model for an external data that is the SMBD possible to solve the limitations. In this reason, the aim of the thesis is to provide a conceptual framework by adapting the RFM model for unstructured information extraction from external data that is requires the advanced capabilities of high technologies. From aforementioned reviews, NLP and OM are the technologies to deal with the characteristics of SMBD and DM is a technique to analyze the information. In this context, the thesis conducts the research applying those technologies in order to re-design and support the model for understanding the customer feedbacks and the products based on their perspectives.

2.2 Natural Language Processing (NLP)

In terms of marketing, customer feedbacks play an important role to advocate the marketing managers for marking the better decisions. This is because it provides the information which can enable to understand their customer opinions. However, the weak point of SMBD is the complexity which is well-known that machine can not directly use it [49], [57]. In this situation, the advanced computing technology is required to automatically process and analyze the data in order to discover the hidden knowledge [58]. The crucial technology to deal with that limitation is Natural Language Processing (NLP) [49], [50].

In general, NLP is the pre-processing steps for structuring the unstructured data [50]. The objective of NLP is to transform the unstructured data and represent into the machine-readable form. It enables the machine to understand the human languages in the same ways as human understanding [49], [58]. In recent years, NLP is popularized in terms of Information Extraction (IE) to mine the relevant information from the abundant feedbacks in the SMBD [49], [58]. The main processes of NLP are sentences or phrases splitting, tokenization, part-of-speech (POS), named entity recognition (NER) [36], [49], [59].

To perform the aforementioned processes easily in a machine, the NLP APIs to support them are available [59]. They are the commercial and non-commercial products purposed with different objectives and supports different platforms. Table 2.2 illustrates the comparison of commonly used NLP APIs by testing on the standard dataset namely, CoNLL [59]. The results obtained from the APIs concern with the accuracy of each process (e.g., Tokenization, POS, NER, and OM). The techniques for processing consist of Machine Learning (ML), Rule-based (RB), and Dictionary-based (DB). They also support the different programming languages.

Table 2.2. The comparison of available NLP APIs [59].

APIs	Programming Language	Methodology				Techniques			Accuracy (F Measure %)
		Token	POS	NER	OM	ML	RB	DB	
NLTK	Python	✓				✓			95
			✓			✓			68
				✓		✓			89
					✓	✓			84
OpenNLP	Java	✓				✓			99
			✓			✓			88
				✓		✓			88
					✓	✓			87
CoreNLP	Java	✓				✓			73
			✓			✓			67
				✓		✓			70
					✓	✓			70
Pattern	Python	✓					✓		42
			✓				✓		35
TweetNLP	Java	✓				✓			98
			✓			✓			84
TwitterNLP	Python	✓						✓	96
			✓					✓	83
				✓				✓	88
					✓			✓	85
TwitIE	Java	✓						✓	89
			✓			✓			82
				✓		✓			77
					✓	✓			76

Table 2.2 shows the OpenNLP from Apache gets the highest accuracy from all of the three processes and NLTK gets the highest accuracy (only different 1% with OpenNLP) in NER. As the results, this thesis is decided to apply the OpenNLP API to develop the system.

2.3 Opinion Mining (OM)

Opinion Mining (OM) is also known as Sentiment Analysis (SA) [57], [60] that is provided the largest set of discovering the emotions in customer feedbacks from the SMBD [38]. Normally, the inputs of OM are the outputs of NLP processes such as Named Entities (NEs). There have basically two approaches of OM which are the corpus-based approach and the machine learning-based approach [38], [57].

2.3.1 Corpus-based sentiment approach

The Corpus-based approach aims to classify the customer feedbacks from the SMBD based on pre-defined keywords known as the bag-of-words model that is the collection of opinion words such as positive words and negative words [16]. The weak point of this approach is that it requires the practitioners or experts to define the important keywords. The general processes of corpus-based approach are as shown in Figure 2.1.

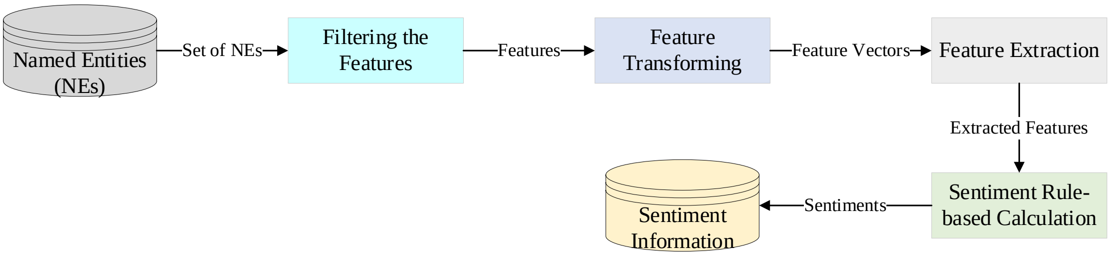


Figure 2.1. The general processes corpus-based OM approach.

Figure 2.1 shows the corpus-based approach that consists of four main components. The first component is filtering the features from NEs to identify the relevant features [38]. Then, the features need to transform into the vector form that are performed in the feature transforming component. This is because in the feature extraction component, it needs to use to compare the similarity, commonly known as cosine similarity, between the sentiment words using Bag-Of-Words model based on a vector space model [33], [40]. Finally, calculating the sentiment-level based on the rule-based algorithm that calculates the count of sentiment words and then produces the level of sentiment.

However, X. Fang and J. Zhan [41] explained the expected difficulties of recognizing the sentiment polarity from customer feedbacks. Generally, the corpus-based approach on sentiment analysis is the Word-count algorithm. It selects and counts the positive and negative corpus from any feedback based on a sentiment dictionary. This algorithm defines the sentiment by counting the numbers of positive and negative transactions. Based on the Word-count algorithm, M. Thelwall and et.al [61] introduced a new algorithm, SentiStrength. Firstly, it assigned a level score to each selected positive and negative corpus from feedbacks, for example, the positive level of the word ‘fine’ is 3, and, for ‘shabby,’ it is -2. Then, it extracted the sentiment based on the total number of assigned values in a feedback.

In recent years, a large and growing body of literature has investigated the hidden knowledge of customer opinions from the SMBD. L. Bing et al. [16] proposed a sentiment analysis from feedbacks using corpus-based approach in order to solve ambiguous textual messages from the contents. The lists of related words were prepared and used for predicting the movement of the stock prices of thirty public companies based on the opinions of customer engagements. The paper showed the average accuracy more than 70%. C. Tang and L. Guo [62] and A. Majumdar and I. Bose [63] proposed an approach to extract the information from SMBD in order to understand the electronic word-of-mount. They also applied a corpus-based algorithm to recognize the information for understanding consumer satisfaction. Moreover, A. R. Alaei and et.al. [44] and H.-C. Chang and C.-Y. Wang [64] represented an approach to extract the marketing information from SMBD. They utilized a word-matching based algorithm using the bag-of-word model to determine meanings of related tokens. D. Vilares and et.al. [35] described an approach to discover the sentiment from SMBD. They practiced a rule-based learning algorithm to determine the customer satisfaction on products and services. Similarly, Sheu and K.-T. Chu [28] proposed an association rules mining system based on a decision tree algorithm to generate the rules for electronic word-of-mouth on SMBD.

Even though many researches had investigated the corpus-based sentiment approach to extract sentiment information, they could not handle the complexities of human languages on SMBD. From that perspective, the aim of this research is to provide a new approach for better handling the complexities of SMBD.

2.3.2 Machine learning (ML) approach

The ML approach classifies the sentiments from SMBD based on a pre-defined model, called text mining. It uses the mathematics and statistics based on the defined model for discovering the knowledge from the textual data. In the text mining approach, Naïve Bayes, Maximum Entropy, and Support Vector Machine (SVM) are the most commonly used [49]. Although the ML approach can give an acceptable accuracy [38], the limitation in terms of requirement of annotated corpora is not easily satisfied, especially for cross-domain. The general processes of ML-based OM approach are as shown in Figure 2.2.

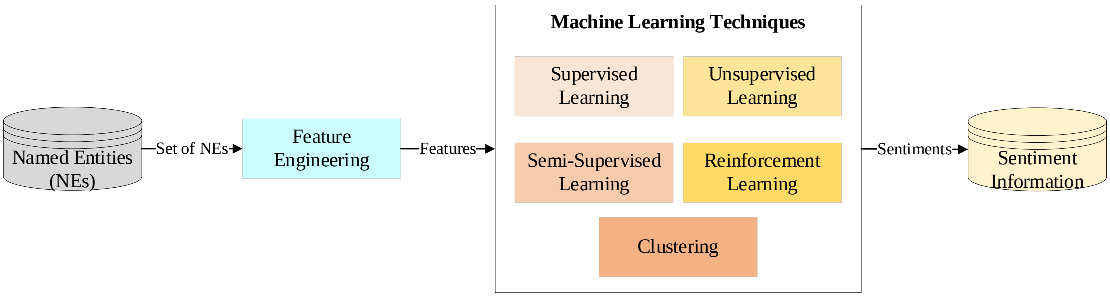


Figure 2.2. The general processes of OM using ML approach.

Figure 2.2 shows the general processes of OM using the ML-based approach that can also be performed to discover the sentiment from SMBD. Firstly, it is the feature engineering which purposed to extract the relevant features from the NEs. In general, it performs by using the feature selection methods [65]. Then, those features are to be delivered to the ML techniques. The ML techniques refer to the algorithms to automatically uncover the hidden sentiment knowledge which consists of supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, and clustering. The aspect sentiment information extraction is a famous technique for the ML-based sentiment approach.

Many researches interested in aspect sentiment information extraction [37], [42]. Aspect sentiment analysis concerns the customer opinions on the desired aspects of the objects such as cost, quality, and services. D. Hazarika and et.al [66] and Y. Ma and et.al [67] proposed an aspect-based sentiment extraction approach. They used a neural network algorithm of the machine learning approach to extract the information based on targeted aspects concerning products and services. N. Majumder and et.al. [43] represented aspect sentiment extraction from SMBD on two different domains: Restaurant and Laptop. They also used a machine learning approach and proved that the aspect-based sentiment can support the marketers to understand their customer satisfactions on the aspects of products and services.

However, the works in the mentioned literatures did not perceive the complexity of human languages in customer feedbacks in extracting the sentiments, and did not apply corpus-based sentiment analysis or aspect sentiments analysis, lacking of combination efforts. E. Cambria [38] explained the benefits of hybrid sentiment approaches (corpus and ML approaches) which can provide more precise information than the individual approach. Therefore, it is presumed that in a combining application, the aspect-based sentiment information extraction can help improve the sentiment information extraction from the SMBD.

2.4 Data Mining (DM) Related with the RFM Model

DM is a well-known technique to mine the goals from raw data using statistical mathematic calculations or rules [68]. DM plays as an important role in data science and artificial intelligence concerning many techniques such as supervised mining, unsupervised mining, and reinforcement mining [69], [70]. Different mining techniques have different perspectives, views, and abilities. For example, supervised mining mines the information based on the prior knowledge that is the training sets involving the information from the past data. Unsupervised mining concerns on data without labels and discovers the information based on the similarity of data features or properties, e.g. clustering [69], [70]. The RFM model mainly focusses on these two types of mining techniques for applying market analysis [1], [4], [5].

In the past decade, the RFM model combining with different data mining techniques have been applied in many researches areas such as customer behaviors segmentation, predicting future customers, in market analytics systems [4], [56]. The well-known and most suitable DM technique for the RFM model is the quintile analysis [4], [5], [52], [56], [71]. It determines to cut-off the levelled scores from the given population into five sets or to represent them in the percentage ranges of 20% per set. The maximum score is five which means the top 20% of given population. The general process of the quintile analysis is shown in Figure 2.3.

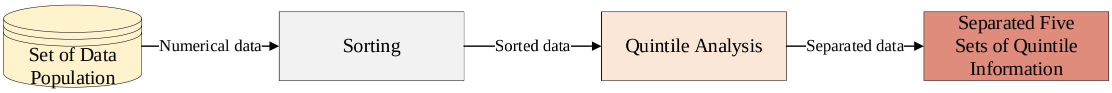


Figure 2.3. The general processes of the quintile analysis.

Figure 2.3 shows how to analyze the data in the quintile analysis. These five sets of the quintile information on the R, F, and M variables can be used to analyze the customer behaviors such as categorizing the similar information of customers' transactions. The information can be understood by the marketers which customers they should target for their products.

The RFM model is a well-known and famous technique in traditional market analysis. However, many researchers proposed the RFM model combining with many DM techniques to apply in real-world applications, for instance, the RFM model with K-means algorithms to grade the customer behaviors [4], [5], [72], the RFM model with multiple mining techniques for supporting to the customer relation managements [3], [7], the RFM model with the pattern mining algorithm to find out the sequential patterns [73], the RFM model of customer behavior segmentation combining in the customer lifetime value (CLV) to find target customers for industries [52], and the RFM model for online game churn player analysis combining with the k-nearest neighbors (k-NN) algorithm [6]. Based on the categories of graded customer behaviors, some researches apply a supervised mining technique to predict the customer actions for future in order to support the marketers for creating their marketing strategies [4].

2.5 Summary

Even though the RFM model is a flavor technique to analyze the customer behaviors in terms of traditional market analysis systems, it has lots of advantages and disadvantages. However, the RFM model is that did not focus on the SMBD. Hence, the main challenge is how to adapt the model for supporting the social market. It also requires advanced techniques, i.e., NLP, OM, and DM to support this adapted model for being applied in SMBD analysis. The NLP is a technique to understand the human languages automatically. Although previous studies demonstrated the recognition of the customer satisfaction from SMBD, they could not handle well on the complexity of human languages on SMBD. The sentences from customers’ feedbacks on SMBD may not be grammatically correct and can also be complex. They often use contrast words in one feedback. The main research gap on this is how to handle the complexity of human languages from SMBD by machine automatically. The OM is another technique to discover human opinions from SMBD. Even though many researches extracted the sentiment information from the SMBD applying many algorithms and techniques (e.g., corpus-based and machine learning approaches), they still lacked of the combination techniques to improve the performance of information extraction. The main challenge on the information extraction is how to hybrid the techniques to improve the performance in terms of sentiment analysis. The DM is the key technique for adapting the RFM model to produce the information and knowledge from SMBD analysis. The quintile analysis is the DM method for applying the RFM model previously used in traditional market analysis to be used in SMBD analytics.

CHAPTER 3
SYSTEM DESIGN AND IMPLEMENTATION

This chapter provides system design and research methodologies of the thesis consisting of three main sections. The definition of the RFC model is as shown in Section 3.1. Then, the overview architecture of the proposed system and detailed processes are presented in Section 3.2. Finally, Section 3.3 is summary.

3.1 The proposed RFC Model

In traditional market analysis, the RFM model is a famous technique to analyze the customer behaviors from direct data such as database and data warehouse. The key strength of the RFM model is that it collects and detects the customer behaviors for a long duration in the traditional market. For instance, the marketing manager can easily know whether the customers are active or passive, considering the recency (R) values. Moreover, the frequency (F) and monetary (M) values can easily help divide the level of customers.

Although the RFM model is famous for customer behavior analysis, it has not previously been applied in social marketing based on SMBD. It also cannot be directly used in the SMBD. Therefore, this research proposes the adaptation form of the RFM model. In this work, the R and F remain the same but the Monetary (M) variable is reformed to the Credit (C) from the customer opinions. This is how the social market analytics is done based on the RFC model. The model aims to categorize products for social marketing based on the customers' feedbacks on SMBD. The RFC refers to Recency (R), Frequency (F), and Credit (C), measuring three main variables to represent the products' status.

- **Recency** refers to the activeness of a product in the current state, defined from the date and time of the last customer feedbacks in a certain period. It indicates which products recently have customers' interests and which products do not during this period. Therefore, the information can enable marketers to deal with the situations more effectively.
- **Frequency** is the number of customer feedbacks on a product in a certain period supporting the assumption that products with more feedbacks

have more customers interests and purchases than products with fewer feedbacks. Once obtaining such information, the marketers may find a way to stop the dissemination that can cause a loss in the market share to get more customers purchasing.

- **Credit** comes from the customers' sentiments on the feedbacks of a product to indicate the direction of market movement such as the annual cycle. It allows the marketers to understand their customers satisfactions and to respond to the critical opinions of the customers' feedbacks. The assumption is that negative feedbacks can sway new customers from buying the product. Also, the marketers should improve their products or services to win more credits.

A product has been reviewed by many customers who are of a group with similar interests in this system. R refers to the latest date of the customer feedbacks, F refers to the number of customers in the product-type group, and C refers to the total sentiment information values of that product in a certain period. These three variables can help interpret and analyze the products to understand the situation changes based on the customer satisfactions on SMBD which support the marketers to make better decisions on their products and services.

3.2 The System Architecture to Apply the RFC Model in SMBD

To automatically process the product categorization from the customer feedbacks based on unstructured data, in this section, we provide the overview of system architecture to discover the knowledge to support the decision-making for tactical marketing. The system architecture is as shown in Figure 3.1. It integrates NLP and Opinion Mining for supporting the RFC model. Applying the three main variables, the quintile analysis and rule-based mining of DM are used to define the level scores on each variable and categorized the products. The products alarming component provides notifications on the changes of products categorizes in time-series information. The architecture consists of five components:

I. **Contests pre-processing based on NLP** refers to the analyzed method for processing the language based on naturally textual representation.

The inputs represent in the unstructured data and semi-structured data. Consequently, this component is categorized into two parts. The first part tags the labels on semi-structured data including date and time, product ID, customer’s preferences and so forth which support the RFC model to find out the R and F values. The second part deals with unstructured data for machine understanding by applying the advanced NLP technique. It consists of the split sentences, tokenization, part of speech, and named entity. Finally, the output of this component is the Named Entities (NEs) to be used in analyzing the C value in the upcoming component.

II. **Opinion Mining** is the process for analyzing the feedback contents using the sets of NEs based on the human emotion to discover and provide the sentiment from their opinions. This process entirely considers the nouns, adjectives, verbs, adverbs to identify the customer perspective with positive or negative sentiments using the proposed contrast dictionary and aspect-based sentiment extraction approaches. The output of this component is the sentiment information of feedbacks and labels of semi-structured data to recognize the RFC values in next component.

III. **RFC variables recognition** recognize the C value by assigning the values on sentiments such as “+1 refers to positive sentiments” and “-1 refers to negative sentiment.” Concurrently, the component recognizes the R and F values from labels of date and time from semi-structured feedbacks. The output of this component is the R, F, and C’s values.

IV. **Product categorization** firstly prepares the variables to define level scores on the R, F, and C values using the quintile analysis. It separates each value of a product into five sets and the maximum number is five. Then, a rule-based mining technique is used to categorize the related products based on the level scores. Due to the quintile analysis, it can be categorized 125 (5x5x5) different product types from three main variables. However, the work in this thesis categorizes only eight types (2x2x2) of products based on two sets of three variables.

V. **Products Alarming** monitors the changes of products categorized from the RFC model in time-series frames. It produces notifications to managers on the product-type fluctuations in times.

According to the overview architecture, NLP is the first component to deal with the unstructured texts. The sentiment information extraction is the second process for recognizing the sentiment in the RFC model. Herein, the detail processes of each component are explained in the next section.

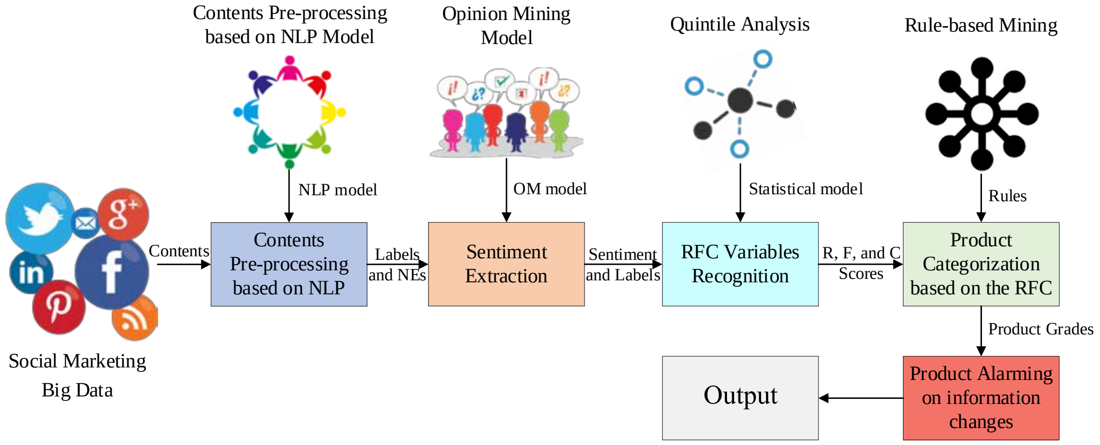

Contents Pre-processing based on NLP Model
Opinion Mining Model
Quintile Analysis
Rule-based Mining
NLP model
OM model
Statistical model
Rules
Contents
Contents Pre-processing based on NLP
Labels and NEs
Sentiment Extraction
Sentiment and Labels
RFC Variables Recognition
R, F, and C Scores
Product Categorization based on the RFC
Product Grades
Social Marketing Big Data
Output
Product Alarming on information changes

Figure 3.1. The proposed architecture to apply the RFC model on SMBD analytics.

3.2.1 Contents pre-processing based on the NLP

The aim of contents pre-processing is to prepare the data into machine readable forms in order to recognize variable information easily. We separate the process of this component consists of into two parts. The first one sets a label on the semi-structured data and unstructured data as shown in Figure 3.2.

Semi-Structured Data						Unstructured Data
Transaction No.	**Product ID**	**Customer ID**	**Helpful**	**Overall**	**Feedback Date**	**Feedback Text**
1	B00002243X	A3F73SC1L Y51OO	4, 4	5	17/04/2014	I needed a set of jumper cables for my new car and these had good reviews and were at a good price.

Figure 3.2. The example of dataset and label on each node.

Figure 3.2 shows an example of the labelled dataset. It divided into two types which set labels on semi-structured data and unstructured data. The labelled semi-structured data can support the RFC model to extract the R and F values. For example, the R value is extracted from the “Feedback Date” and the F value comes from the total “Transaction” number. The “Product ID” shows the actual product focused in the categorization and the “Customer ID” refers to the consumer of the product. The unstructured data is the input for the second process using NLP techniques for recognizing the sentiment and C value.

In general, NLP includes the pre-processing steps for structuring the unstructured data. The objective of NLP is to transform the unstructured data into a machine-readable form. In recent years, NLP is popularized in terms of contents pre-processing to mine the relevant information from abundant feedbacks in the SMBD [49]. The main components of the NLP concept are sentences or phrases splitting, tokenization, normalization, part-of-speech (POS), and named entity recognition (NER). The architecture of NLP is as shown in Figure 3.3.

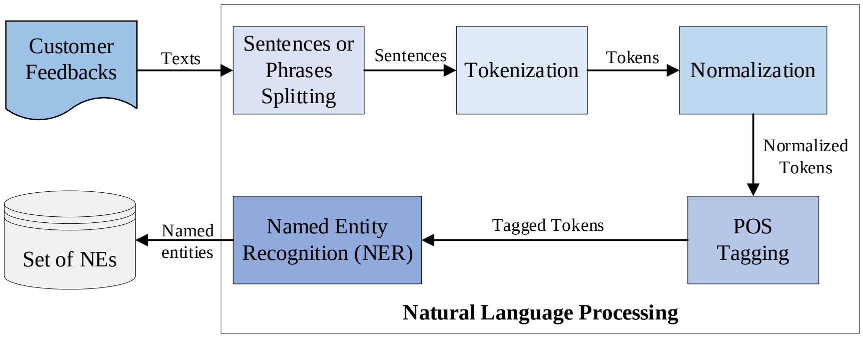


Figure 3.3. The architecture of Natural Language Processing.

Figure 3.3 shows that the raw data of customer feedbacks from the SMBD is the input for NLP processes. The data is to be read from the storage and forwarded to the Sentences or Phrases Splitting component.

1. The splitting sentences or phrases process splits the sentences from the feedbacks to improve the performance of the Opinion Mining approach.
2. The tokenization component is for splitting the string from original sentences into words called tokens.
3. The third component is normalization. This component is removing noises (e.g., stop-words, articles, and symbols) and correcting

misspelled words (e.g., misspelling, abbreviation, and slang) to improve the quality of the sets of tokens.

4. Fourth, the POS component annotates or tags the relevant tokens by using syntactic language statements such as noun, verb, adjective, adverb, and so forth. It calls the tagged tokens.
5. The last component is Named Entity Recognition (NER). This component helps identify the names of tagged tokens which usually are the names of persons, products, places, organizations, and so on. The outputs of this components are Named Entities (NEs).

3.2.1.1 Sentences or Phrases splitting

Customer feedbacks on SMBD are normally represented in unstructured texts. These texts need to be separated into sentences or phrases to improve the performance of information recognition. Firstly, the sentences or phrases are split based on punctuations, e.g., full stop (.), exclamation mark (!), and question mark (?). Secondly, the sentences or phrases are split based on the conjunctions, e.g., ‘and’, ‘what’, and ‘which’. The processes use the regular expression method. An example of the process is shown in Table 3.1.

Table 3.1. The sentences or phrases splitting example.

Original Data String	Split Sentences or Phrases
this product does what it is supposed to do but I am not real happy with the quality of construction.	this product does
	what it is supposed to do
	but I am not real happy with the quality of construction.

3.2.1.2 Tokenization

The sentences or phrases strings are not possible to use in machine to deep information recognition. The objective of the tokenization is to parse the split sentences or phrases into words. The separated words calls the sets of tokens that we use an opensource software, Apache OpenNLP [74]. It processes the data based on a machine learning technology, the maximum entropy, using a training set. An example of the tokenization process is shown in Table 3.2.

Table 3.2. Tokenization process example.

Split Sentences	Tokenization
this product does	this | product | does |
what it is supposed to do	what | it | is | supposed | to | do |
but I am not real happy with the quality of construction.	but | I | am | not | real | happy | with | the | quality | of | construction | . |

3.2.1.3 Normalization

The obstacles of information extraction are noises that need to remove for improving performance. In terms of NLP, the noises are stop words (e.g., articles and symbols) and anonymous (e.g., slang and misspelling). The normalization aims to remove them to improve the sets of tokens quality. In this point, the stop-words dictionary is applied to analyze and remove the stop words from the token sets. An opensource software, Apache Lucene [75], is used in combination with the N-Gram algorithm [76] to correct the misspelling words. An example of this process is shown in Table 3.3.

Table 3.3. Normalization process example.

Tokenization	Clean Tokens
this | product | does |	product |
what | it | is | supposed | to | do |	supposed |
but | I | am | not real happy | with | the | quality | of | construction | . |	but | not | real | happy | quality | construction |

3.2.1.4 Part-of-speech (POS)

The machine also needs to understand the human languages of SMBD as the same ways with human understanding. The aim of POS is to annotate the grammatic meaning to each clean token. The Apache OpenNLP [74] is applied in this process which tagged the token that is noun, verb, adjective, or so on. An example of POS tagging is shown in Table 3.4.

Table 3.4. POS tagging process example.

Clean Tokens	POS Tagging
product |	<Noun> product </Noun>
supposed |	<Adj> supposed </Adj>
but | not | real | happy | quality | construction |	<Conjunction> but </Conjunction>, <Adverb> not </Adverb>, <Adj> real </Adj>, <Adj> happy </Adj>, <Noun> quality </Noun>, <Noun> construction </Noun>

3.2.1.5 Named entity recognition (NER)

Named Entity Recognition (NER) is a principle concept in artificial intelligence for annotating the meaning of an object in a machine-readable form [77]. NER is also known as a sub-component in information extraction to enable the machine to understand a human natural language. The objective of NER in this work is to describe the semantic meanings to the sets of tokens. The maximum entropy algorithm is used by applying the Apache OpenNLP API [74]. An example of the NER process is shown in Table 3.5.

Table 3.5. NER process example.

Clean Tokens	NER
product |	<Product> product </Product>
supposed |	supposed
but | not | real | happy | quality | construction |	<Contrast> but </Contrast>, not, real, happy, <Quality> quality </Quality>, construction

3.2.2 Sentiment extraction

Sentiment extraction is a main principle involving in this research. The credit (C) information of the RFC is mainly depended on the performance of sentiment information extraction. From that perspective, sentiment extraction is play an important role of the RFC model to apply in SMBD analytics. The thesis proposes and uses a new multiple levels sentiment extraction algorithm, called contrast dictionary. It can handle the complexity of human languages usage on SMBD more than well-known dictionary-based algorithms, Word-count and SentiStrength.

A new hybrid approach is also proposed to extract sentiment that called aspect-based sentiment extraction. It processes based on the fusing results of the contrast dictionary algorithm and the machine-learning-based aspect-level sentiment analysis. The architecture of sentiment extraction is as shown in Figure 3.4.

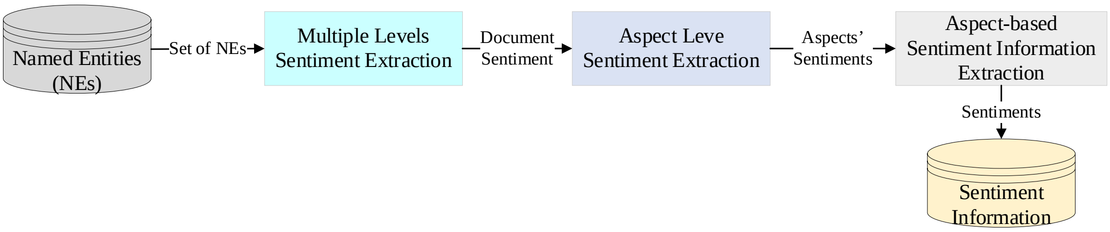

Named Entities (NEs)
Set of NEs
Multiple Levels Sentiment Extraction
Document Sentiment
Aspect Leve Sentiment Extraction
Aspects' Sentiments
Aspect-based Sentiment Information Extraction
Sentiments
Sentiment Information

Figure 3.4. The architecture of sentiment extraction processes.

Figure 3.4 shows the system architecture to extraction the sentiment information from SMBD. The inputs come from the outputs of the NLP processes. It includes three main process to define the final sentiment information of a feedback.

3.2.2.1 Multiple Levels Sentiment Extraction (Contrast Dictionary)

In this component, a new algorithm, called the contrast dictionary is proposed. It is applied on multiple levels: word, sentence or phrase, contrast, and document, to extract the sentiment information from customer feedbacks as shown in Figure 3.5.

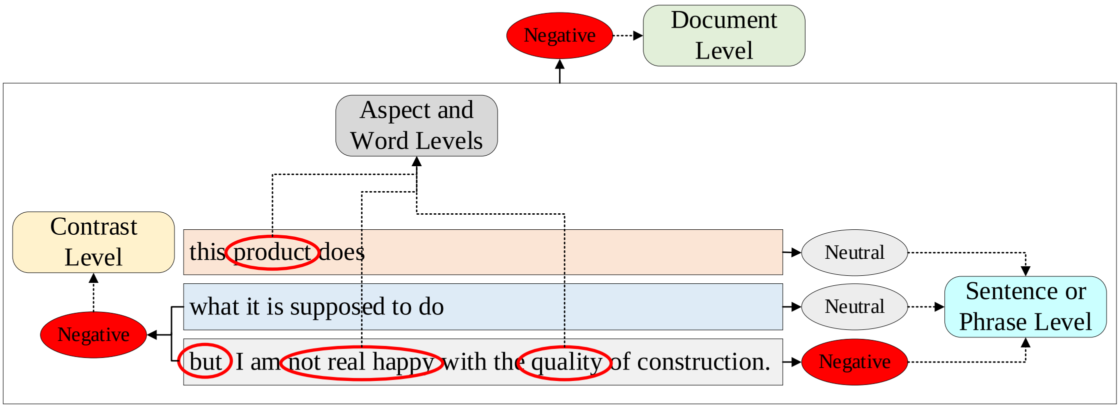


Figure 3.5. An example of multi-level sentiment information.

Figure 3.5 shows the example of multi-level sentiment information on a feedback. The sentiment information on the words or phrases are the word-level sentiment. This is somehow the aspect-level sentiment information if the word level sentiment combines with the aspect features. The sentence-level or phrase-level represents the sentiment information that the accumulation of the "Positive" and "Negative" on each sentence or phrase. The modification sentiment of a contrast word on compound or complex sentences is called the contrast-level sentiment. Finally, the document-level sentiment is the summarization of aforementioned sentiments. The processes of the contrast dictionary algorithm on the multiple levels are as shown in follows.

3.2.2.1.1 Word and sentence-or-phrase levels sentiment extraction

The objective of word and sentence-or-phrase levels of sentiment extraction is to improve the accuracy. The word-level sentiment extraction uses the word count algorithm to detect sentiment words (positive, negative, and negation words) including contrast words in each sentence. It defines the sentiment of any sentence or phrase based on the "positive" and "negative" counts as shown in Algorithm 1.

Algorithm 1 demonstrates the processes of word and sentence-or-phrase levels sentiment extraction. The input is the sentences from a customer feedback. For the word-level sentiment extraction, it uses the dictionary-based algorithm to select the positive, negative, and negation words. Then, the negation word is checked, and the sentiment polarity will be modified accordingly. The example of word-level sentiment extraction is shown in Figure 3.6.

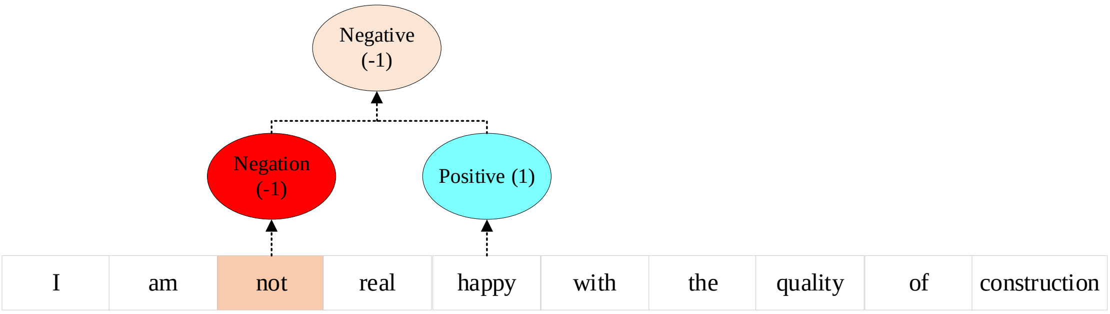


Figure 3.6. An example of word-level sentiment extraction.

The sentence-or phrase-level sentiment is extract based on the word-level sentiments. For example, the sentence will be declared 'positive' if the number of collected ‘positive’ results is greater than that of ‘negative’. The example of sentence-level or phrase-level sentiment extraction process is shown in Figure 3.7. Moreover, Algorithm 1 extracts the contrast word of each sentence. The extracted sentences’ sentiments and contrasts are kept in a list to be sent to the next steps.

Algorithm 1: Word and phrase-or-sentence levels’ sentiments analysis

Input: Sentences

```
A = {Pos_List, Neg_List}
B = {Negation _List} // For example, no and not
C = {Contrast_List}
O = Token (Sentences)
D = Remove_StopWords (O)
FOR k ϵ [0 to D.Size()] DO
   FOR l ϵ [0 to Pos_List.Size()] DO
      IF (D [k] = Pos_List [l]) THEN
         Count_pos++, sentiment = “1”
   FOR n ϵ [0 to Neg_List.Size()] DO
      IF (D [k] = Nega_List [n]) THEN
         Count_neg++, sentiment = “-1”
   FOR m ϵ [0 to Negation_List.Size()] DO
      IF (D [k] = Negation_List [m]) THEN
         neg_word = “Negation”
   IF (neg_word = “Negation” && sentiment = “1”) THEN
      Count_pos--, Count_neg++, sentiment = “-1”
   IF (neg_word = “Negation” && sentiment = “-1”) THEN
      Count_pos++, Count_neg--, sentiment = “1”
   FOR l ϵ [0 to Contrast_List.Size()] DO
      IF (D [k] = Contrast_List [l]) THEN
         Contrast = “Contrast”
   IF (Count_pos > Count_neg) THEN
      Sentence_Sentiment = “Positive”
   ELSE IF (Count_pos < Count_neg) THEN
      Sentence_Sentiment = “Negative”
   ELSE (Sentence_Sentiment = “Neutral”)
   WordSentenceLevelContrastSentiment_List.add (Contrast, Sentiment)
RETURN WordSentenceLevelContrastSentiment_List
```

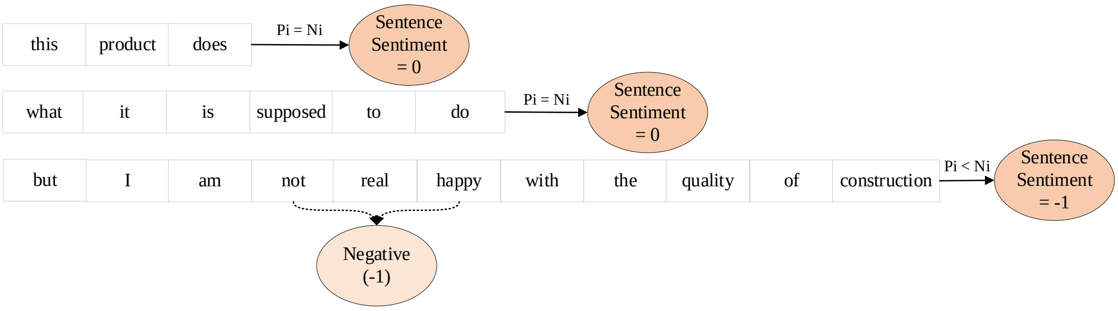


Figure 3.7. An example of sentence-or phrase-level sentiment extraction.

Algorithm 2: Contrast and document levels' sentiments analysis

Input: WordSentenceLevelContrastSentiment_List

```
B [] = WordSentenceLevelContrastSentiment_List.toArray()
FOR n ϵ [0 to B.Size()] DO
  IF (B [n] = "Contrast") THEN
    IF (B [n-1] = "Positive" && B [n+1] = "Positive") THEN
      Contrast_check = "Pos"
    ELSE IF (B [n-1] = "Positive" && B [n+1] = "Negative") THEN
      Contrast_check = "Neg"
    ELSE IF (B [n-1] = "Negative" && B [n+1] = "Negative") THEN
      Contrast_check = "Neg"
    ELSE IF (B [n-1] = "Negative" && B [n+1] = "Positive") THEN
      Contrast_check = "Pos"
    ELSE Contrast_check = "Neutral"
    B [n] = Contrast_check
    B [n-1] = null
    B [n+1] = null
  IF (B [n] != null && B [n] != "Neutral") THEN
    Contrast_Level_List.add (n)
Pos_count = Count ("Positive")
Neg_count = Count ("Negative")
Positive_average = 100 * (Positive_count / (Positive_count + Negative_count)) %
IF (Positive_average >= 50%) THEN
  Sentiment = "Positive"
ELSE
  Sentiment = "Negative"
RETURN Sentiment
```

3.2.2.1.2 Contrast and document levels sentiment extraction

The contrast-level also uses the dictionary-based approach. D. Vilares and et.al explained the sentiment of a sentence with a contrast that the sentiment word of a contrast word can encompass and cover the sentiment word from the front of contrast [35]. For instance, the sentence "I don't like this car, but it has attraction that I really like." is a positive sentence because of the contrast word 'but.' The positive sentiment behind 'but' can override the negative sentiment of the phrase before 'but.' The word 'but' is a contrast word to compliment the entire feedback sentiment. This insight is applied to determine the overall sentiment of a sentence. The contrast and document levels sentiment extraction are shown in Algorithm 2.

The input is the list from the output of Word and phrase-or-sentence levels sentiment analysis. In this algorithm, the contrast word from the tokens (sentences or phrases) is checked using the contrast dictionary. If a sentence uses a contrast word, the algorithm will check the adjacent (prior and subsequent), and current sentences or phrases. The nested sentences or phrases will be combined into one context based on the contrast word giving a fused sentiment result. The example of the contrast-level sentiment analysis is shown in Figure 3.8.

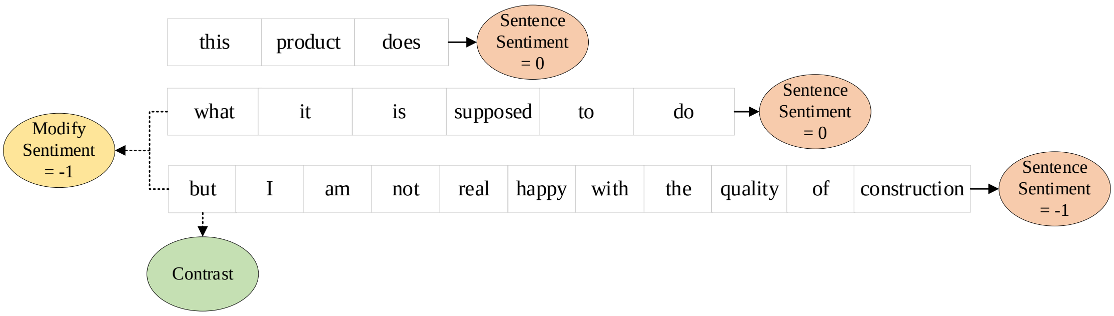


Figure 3.8. An example of contrast-level sentiment extraction.

The document-level sentiment analysis is the final process of the contrast dictionary algorithm to provide the final sentiment result of the feedback. It counts the positive and negative sentences or phrases and calculates the percentage of the positive and negative sentiments or phrases in order to define the sentiment of the feedback as a whole. In the case study, a feedback's sentiment is declared 'Positive' if the positive sentiments' percentage equals to or more than 50%. The percentage lower than 50%

indicates a negative one. This positive to negative ratio empirically gets better accuracy than the others, e.g., 55% and 45%, 60% and 40%, and 45% and 55%. The example of document-level sentiment extraction is shown in Figure 3.9.

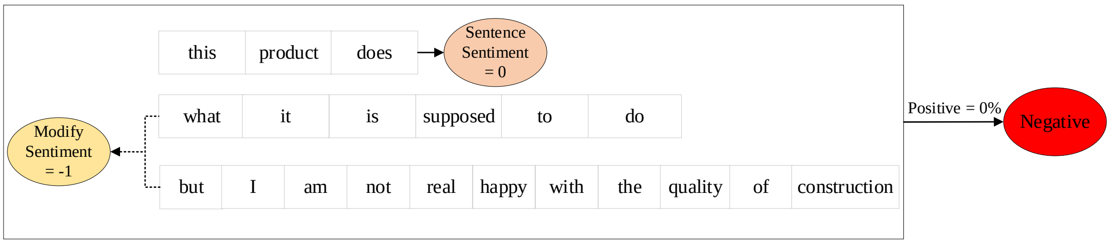


Figure 3.9. An example of document-level sentiment extraction.

3.2.2.2 Aspect-level Sentiment Extraction

Aspect-level sentiment extracts the information based on the desired aspects. This research applies the Named Entity Recognition (NER) technique to extract the aspect sentiments. The objective of NER in this work is to describe the semantic meanings to the sets of tokens using the maximum entropy in machine learning that applies with the Apache OpenNLP [74] combining with the N-gram theory [76]. This component extracts the aspect-level sentiment from feedbacks.

Table 3.6. The targeted aspects on tested products.

No.	Product Name	Targeted Aspect
1.	Car Duster	**Product, Durability, Quality, Notification, Recommendation, Cost, Usage, and Applying Period**
2.	Funnel	
3.	Absorber Synthetic Drying Chamois	
4.	Wheel Chock	
5.	Jack Stands	

Even though not all transactions include aspects, various aspects can be extracted from some feedbacks depending on the types of products. The customer feedbacks on different products involve different aspects. For example, feedbacks on cars normally consist of car designs and services aspects, but cable products do not include similar aspects. Herein, this research focusses on five Amazon’s Automotive products and targets eight types of aspects as shown in Table 3.6. The aspect sentiment features is extracted from 305 records on those products and train models for each aspect using the Apache OpenNLP [74]. The aspect models are tested on 100 different feedbacks. Example features are shown in Table 3.7.

Table 3.7. Examples of aspect sentiment features.

No.	Aspect	Example Features
1.	**Product**	grade product, nice product, terrible product
2.	**Durability**	using for years, durability, easily broken
3.	**Quality**	not happy quality, good quality, grade quality
4.	**Notification**	just make sure, be careful, important, suggest
5.	**Recommendation**	strongly recommend, not recommend, do not use
6.	**Cost**	worth to pay, cheaper, expensive
7.	**Usage**	easy to apply, easy to use, difficult to use
8.	**Applying Period**	have to wait two days, take time to apply, finish in a few minutes

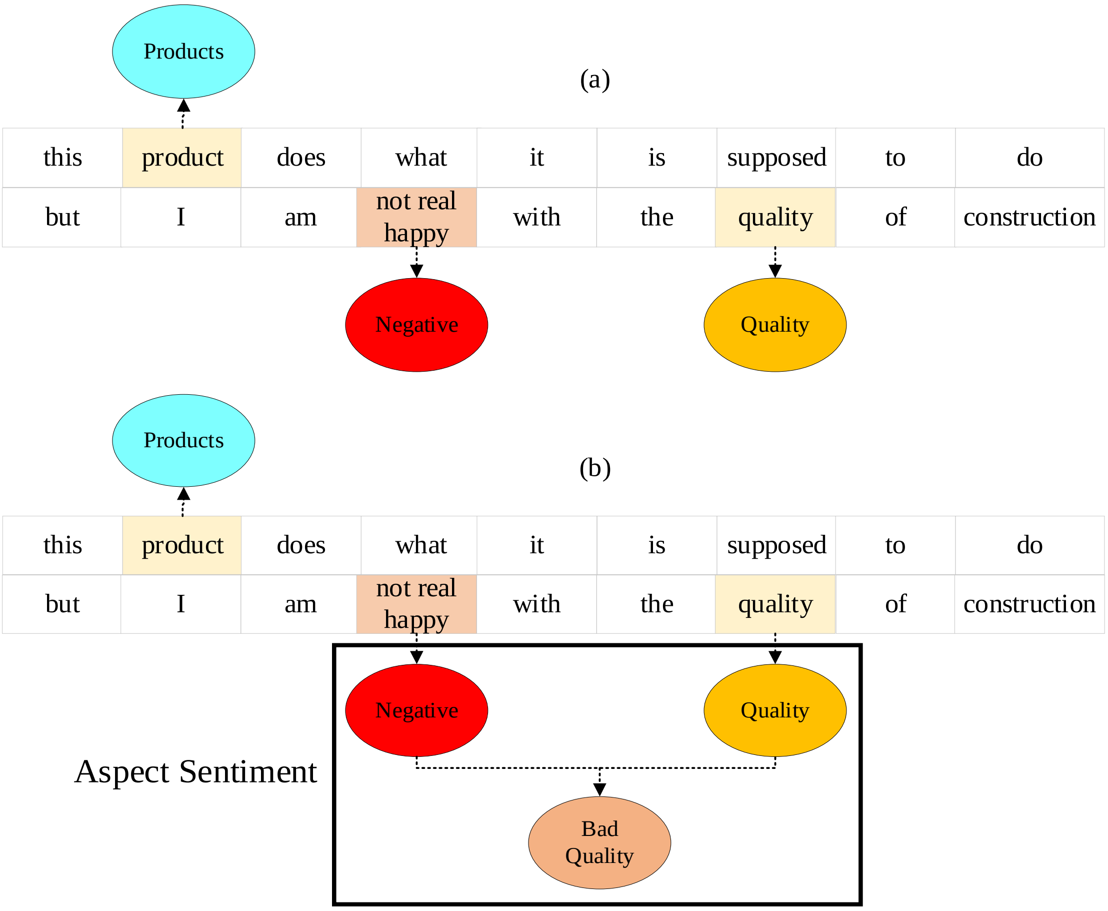


Figure 3.10 (a). An example how NER annotates the meanings to target tokens.

Figure 3.10 (b). An example of aspect-level sentiment information extraction.

These corpuses are collected from the Table 5 and create models to extract aspect-level sentiments determining the "**Product**" (good/bad), "**Durability**" (short/long), "**Quality**" (good/bad), "**Notification**," "**Recommendation**," "**Cost**" (effective/ineffective), "**Usage**" (easy/difficult), and "**Applying Period**" (short/long). The features are transformed into vector forms in each training model for annotating or extracting the aspect features of testing data. An example of annotation named to the target tokens for extracting aspect sentiments is shown in Figure 3.10.

Figure 3.10 (a) shows an example how the sets of tokens are named by using our features' models applying the NER technique of the Apache OpenNLP API with the N-gram. For example, the "product" token is the "**Product**" aspect that the customer mentioned in feedbacks. The phrase "not real happy" has a negative sentiment, and "quality" implies the "**Quality**" aspect. Figure 3.10 (b) represents an example of aspect-level sentiment extraction. In this example, the customer mentioned the aspects of "**Product**" and "**Quality**", which can indicate that the "**Quality**" is bad.

Algorithm 3: Aspect-based Sentiment Extraction

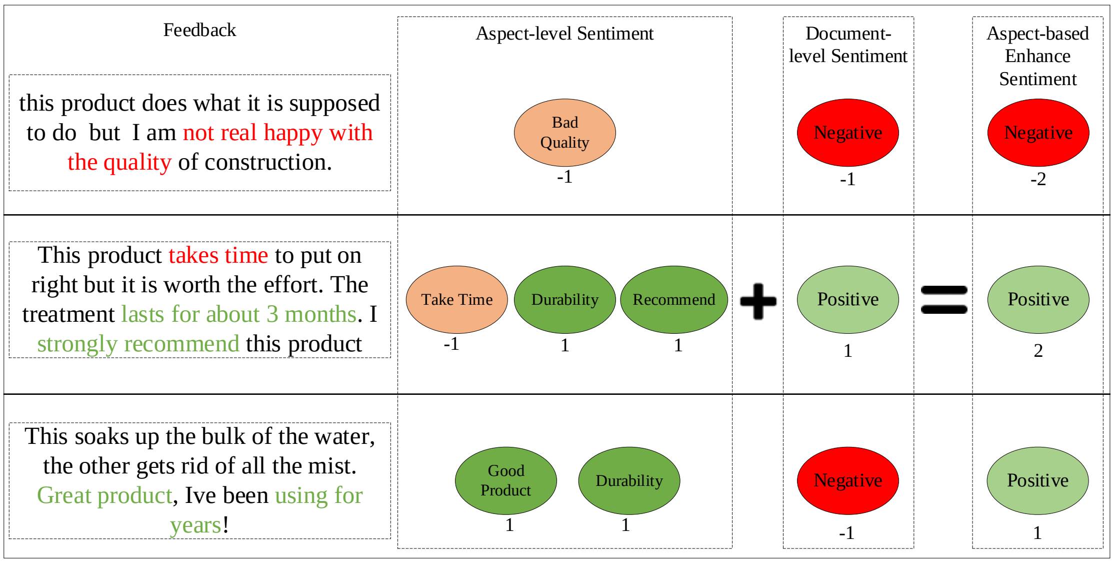
Input: Aspect-level sentiments and the document-level sentiments

```
1. Detect the feedback including aspects
2. Assign the level values on each sentiment
3. Accumulate all sentiments
4. IF the total value is positive
5.     Assign the feedback's sentiment to "Positive"
6. ELSE IF the total value is negative
7.     Assign the feedback's sentiment to "Negative"
8. ELSE
9.     Invert the feedback's document-level sentiment
```

3.2.2.3 Aspect-based Sentiment Information Extraction

The aim of the aspect-based sentiment information extraction is to improve the performance of sentiment information extraction from feedbacks. The multiple levels' sentiments have many mistaken results from feedbacks. To improve the mis-results, the aspect-based sentiment provides to fuse the sentiments from the multi-level and aspect-level sentiments extraction. The methodology of the proposed aspect-based enhancing sentiment extraction is shown in Algorithm 3.

Algorithm 3 shows the aspect-based enhancing sentiment extraction. It assigns values to the document-level sentiments of the multiple levels' sentiment and the aspect-level sentiments and then accumulate all sentiment values. The feedback is defined 'positive' for the accumulated result greater than zero and 'negative' if less than zero. As a result, the final sentiment can be inverted. The example of aspect-based enhance sentiment extraction is shown in Figure 3.11. The output sentiment is the final sentiment of a feedback to recognize the C value.



this product does what it is supposed to do but I am not real happy with the quality of construction.

This product takes time to put on right but it is worth the effort. The treatment lasts for about 3 months. I strongly recommend this product

This soaks up the bulk of the water, the other gets rid of all the mist. Great product, Ive been using for years!

Figure 3.11. An example of aspect-based enhancing sentiment extraction.

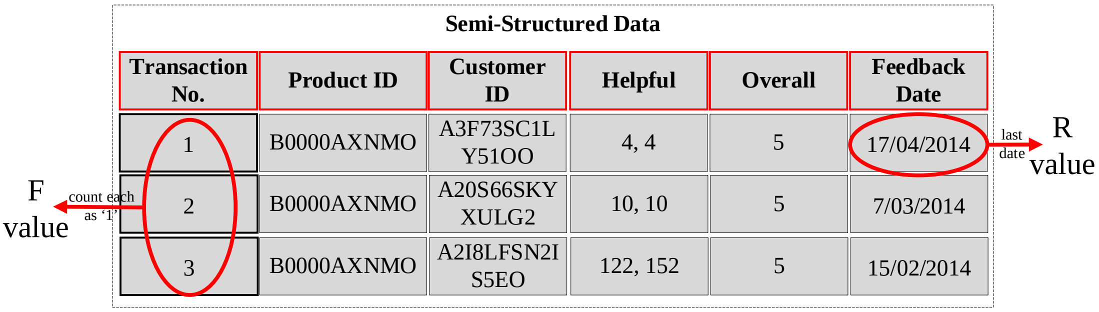
3.2.3 RFC variables recognition

This process recognizes the RFC values from the outputs of previous processes. For instance, the R value is the last "Feedback Date" of a product and the F value is the total number of transactions feedbacks commenting on a product which is recognized from the labelled semi-structure data, and the C value is the accumulation of sentiments from the OM component. The values are assigned on each sentiment, 1 is the value of a positive feedback and -1 is the value of a negative feedback. The methodology to recognize the R, F, and C values is as shown in the Algorithm 4. An example of the R and F values recognition from semi-structured data is shown in Figure 3.12.

Algorithm 4: R, F, and C variables recognition

Input: The outputs of NLP and OM

```
1. Define a certain period
2. Assign the lasted date of the period
3. Select the value of the last feedback date
4. The R value is the distance of the feedback date and the period date
5. The F value is the total number of transactions on a product
6. The C value is the total sum of sentiment values
```



Transaction No.	Product ID	Customer ID	Helpful	Overall	Feedback Date
1	B0000AXNMO	A3F73SC1LY51OO	4, 4	5	17/04/2014
2	B0000AXNMO	A20S66SKYXULG2	10, 10	5	7/03/2014
3	B0000AXNMO	A2I8LFSN2IS5EO	122, 152	5	15/02/2014

Figure 3.12. An example of the R and F values from feedbacks on a product.

Figure 3.12 shows an example of variable recognition from semi-structured data based on the "Product ID". The R value is the last feedback date and the F value is the total number of feedbacks on a product in a certain period. Based on this recognition, an example of the R, F, and C values is shown in Table 3.8.

Table 3.8 shows an example dataset for the RFC model recognized from the SMBD. The Recency (R) value represents the activeness of a product information how many days ago the product got the last comment. The Frequency (F) value shows the population of customer interested on a product in this period. The last variable is the Credit (C) value that describes the accumulation of customers’ satisfactions in the period.

Table 3.8. An example of aspect sentiment features on each aspect.

Product ID (PID)	Recency (R)	Frequency (F)	Credit (C)
B00029WYVW (P1)	80 (day ago)	3	3
B00008RW9U (P2)	19 (day ago)	7	5
B0000AXRH5 (P3)	44 (day ago)	9	5
B0002JMIEK (P4)	9 (day ago)	6	6
B0000AXNMO (P5)	55 (day ago)	9	7
B0000AY9W6 (P6)	58 (day ago)	15	7
B00026Z3DQ (P7)	34 (day ago)	11	7
B0002F9YHI (P8)	28 (day ago)	12	8
B0000AY3X0 (P9)	42 (day ago)	14	12
B00029WYEY (P10)	65 (day ago)	21	13

3.2.4 Product categorization based on the RFC model

After recognizing the variable values, the score calculation and definition for the RFC integration model are needed. The statistical solution known as the quintile analysis is used. How to calculate the scores from the RFC model is illustrated in Equation (1). It determines to cut-off the levelled scores from the given population into five sets or to represent them in the percentage ranges of 20% per set based on the ascending or descending order. The maximum score is five which means the top 20% of the given population. An example calculation using the quintile equation on the example dataset of credit (C) values from Table 3.8 is shown in Table 3.9.

$$\mathrm{R}_5 = 5 \times \frac{Position + \left(\frac{1}{2}\right) Equal}{Total} \tag{1}$$

Where R_5 refers to the set of population which represents each 20% of all population, *“Position”* refers to the position number from the order, *“Equal”* refers to the total number of the equal values, *“Total”* refers to the total number of the transactions of the dataset. An example of the quintile analysis on the C values is shown in Table 3.9.

Table 3.9. Example levelled scores of the RFC model based on the quintile analysis.

Sorting No.	Product No.	C value	The calculated score using the quintile equation	The score	Rank percentage
1	P1	3	0.75 ≈ 1	1st	1-20%
2	P2	5	1.5 ≈ 2	2nd	21-40%
3	P3	5	2		
4	P4	6	2.25 ≈ 2		
5	P5	7	3.25 ≈ 3	3rd	41-60%
6	P6	7	3.75 ≈ 4	4th	61-80%
7	P7	7	4.25 ≈ 4		
8	P8	8	4.25 ≈ 4		
9	P9	12	4.75 ≈ 5	5th	81-100%
10	P10	13	5.25 ≈ 5		

Table 3.9 represents the levelled scores of the C value by using the quintile analysis. Firstly, it sorts the population of C values by the ascending order. For instance, the maximum value of all product groups is 13 and the calculated score is 5.25 so that the levelled score is 5. The minimum value is 3 and the level score can be 1 as the calculated score is 0.75. In case of 6, it can define 2.25 ≈ 2, 8 can define 4.25 ≈ 4, the first sorting value 7 can define 3.25 ≈ 3 and so forth.

The quintile analysis can categorize 125 (5x5x5) different product types based on the three main variables (R, F, and C). However, this work categorizes only eight types of products applying a rule-mining technique. Although the quintile analysis allows to set five different classes, we exploit only two sets in this case study. The first three scores from the quintile analysis are defined as “**Down**” and 4 to 5 are defined as “**Up**.” The up and down values depend on the users’ definitions. An example of up and down states based on the RFC scores from Table 3.9 is shown in Table 3.10.

Table 3.10. An example of up and down variables based on the RFC scores from Table 3.9.

Product No.	C value	The quintile score	The set
P1	3	1st	Down (↓)
P2	5	2nd	
P3	5		
P4	6		
P5	7	3rd	
P6	7	4th	Up (↑)
P7	7		
P8	8		
P9	12	5th	
P10	13		

Table 3.11. The RFC pattern sets for clustering products.

Cluster No.	R	F	C	RFC Pattern
1.	↓	↓	↓	R↓F↓C↓
2.	↓	↓	↑	R↓F↓C↑
3.	↓	↑	↓	R↓F↑C↓
4.	↓	↑	↑	R↓F↑C↑
5.	↑	↓	↓	R↑F↓C↓
6.	↑	↓	↑	R↑F↓C↑
7.	↑	↑	↓	R↑F↑C↓
8.	↑	↑	↑	R↑F↑C↑

Based on the up and down states of the R, F, and C values, patterns are set to define the rules to categorize the products as shown in Table 7. It categorizes the products into eight different clusters due to the two scores (2x2x2) on three variables. It shows that a product in Cluster 8 is the best product and the Cluster 1 is the poorest product.

Applying the RFC patterns or the cluster numbers, the marketers can understand their products from the feedbacks easily as below.

- R↑F↑C↑ – is the best product that has recent customer feedbacks, a high frequency and most feedbacks are positive.
- R↑F↑C↓ – refers to the product that has recent customer feedbacks, a high frequency but a low credit which means that there are a lot of negative opinions.
- R↑F↓C↑ – refers to the product that has recent customer feedbacks, a low frequency and a high credit which means that most customer opinions are positive.
- R↑F↓C↓ – refers to the product that has recent customer feedbacks, a low frequency and a low credit.
- R↓F↑C↑ – refers to the product that does not have recent customer feedbacks but has a high frequency and a high credit. This means that the product was popular in the past.
- R↓F↑C↓ – refers to the product that does not have recent customer feedbacks but has a high frequency and a low credit which means that the product has a lot of negative customer opinions in the past.
- R↓F↓C↑ – refers to the product that does not have recent customer feedbacks but has a low frequency and a high credit. The product had a few customers with positive opinions in the past.
- R↓F↓C↓ – refers to the poor product that does not have recent customer feedbacks and has a low frequency and a low credit. In this situation, the marketers should promote to get more customers' interests.

These patterns can be transformed to be rules to categorize the products in the system which are:

I. **Cluster 1:** if Recency = Down, Frequency = Down, and Credit = Down

II. **Cluster 2:** if Recency = Down, Frequency = Down, and Credit = Up

III. **Cluster 3:** if Recency = Down, Frequency = Up, and Credit = Down

IV. **Cluster 4:** if Recency = Down, Frequency = Up, and Credit = Up

V. **Cluster 5:** if Recency = Up, Frequency = Down, and Credit = Down

VI. **Cluster 6:** if Recency = Up, Frequency = Down, and Credit = Up

VII. **Cluster 7:** if Recency = Up, Frequency = Up, and Credit = Down

VIII. **Cluster 8:** if Recency = Up, Frequency = Up, and Credit = Up

3.2.5 Products Alarming on information changes

This Products Alarming component provides monitoring and alarming on the RFC information and product category changes. The alarming techniques can be adjusted depending on the users' requirements. The examples of alarming rules based on the RFC moving patterns to notify the managers on products situations are shown in Algorithms 5 and 6.

Algorithm 5: Alarming on the R, F, and C Information Changes

Input: The R, F, and C values

```
Monitoring the information changes
If the R value gap becomes big
    Notify "Not Recent Feedbacks"
If the accumulated F value becomes stable
    Notify "Currently, No Customer Interests"
If the accumulated C value becomes lower
    Notify "Currently, Many Negative Opinions"
If the distance between F and C values become big for a certain period
    Notify "Recently, Many Negative Opinions"
…..
END
```

Algorithm 5 shows examples of alarming rules on the R, F, and C information changes in a certain period. The users can edit their requirements in order to monitor the information changes on SMBD. For example, when the R value gap becomes big, it means that the product has not recent feedbacks and the alarming system should send a notification on it. By nature, the accumulated F value always goes up. If it stops growing up for a period, the system will notify that the product has not recent customer interests or there are no feedbacks. However, the accumulated C values depends on the sentiment information which can be dynamically fluctuated in trends. Therefore, if this value is going down, the system should alarm that this product increasingly has many negative opinions. Moreover, Algorithm 6 represents examples of alarming rules based on the changes of product categories. Even though the system alarms on the R, F, and C information changes, the alarming on the product changes are also required for a confirmation.

Algorithm 6: Alarming on Product Information Changes

Input: The categorizes from the RFC model

```
1. Monitoring the information changes
2. If Cluster changed from 8 to 7:
3.     Notify “Currently, Many Negative Opinions”
4. If Cluster changed from 8-7 to 6-5:
5.     Notify “Currently, Lower Customer Interests”
6. If Cluster changed from 8-5 to 4-1:
7.     Notify “Not Recent Feedbacks”
8. …..
9. END
```

3.3 Summary

This section provides the detail processes of the research methodologies to categorize the products based on the RFC model. The advance techniques of NLP are transformed the unstructured data of human languages into the machine-readable forms. Multi-level sentiment (Contrast Dictionary) is a process to extract the sentiment and aspect-level sentiment extracts the aspects’ sentiments of a feedback. Moreover, the fuse results of these two types of sentiments are combined and compared to improve the performance of sentiment information extraction, called the aspect-based sentiment information extraction.

The system recognizes the R, F, and C variables information from the aforementioned processes based on the product IDs. These variables are applied the quintile analysis algorithm of a data mining technique which separates the population of each variable into five different sets. The rule-based mining approach categorizes the products based on the comparison of quintile information patterns. Moreover, this chapter explains how to use the RFC model in an alarming system to notify the information changes on products.

The Aspect-based algorithm is to be tested on the 305 customer feedbacks of ten products from the Amazon’s Automotive products. It includes 257 “Positive” and 48 “Negative” records dated starting from the 1/10/2017 to 31/12/2018. The program has been implemented using the Java programming language and the MySQL database. The important codes of the system are listed in Appendix C.

CHAPTER 4
RESULTS AND DISCUSSIONS

This chapter provides the experimental results and discussions on the thesis. Section 4.1 evaluates the performance of the Contrast Dictionary of sentiment analysis algorithm comparing to the standard algorithms. The proposed approach of hybrid sentiment information extraction is discussed in the Section 4.2. The comparison of this work with the previous works on the SMBD related to the R, F, and C variables is shown in the Section 4.3. Moreover, it explains the product categorization applying the RFC model. Finally, the summary of the chapter is described in the Section 4.4.

4.1 The Performance Measurement of the Contrast Dictionary Algorithm

The capability of the RFC model applying in the SMBD mainly depends on the quality of the sentiment information recognition algorithm. This thesis proposes a new sentiment extraction algorithm, called the Contrast Dictionary. The results of the algorithm are to be compared against the results of the two well-known algorithms, Word-count and SentiStrength. Only sentiment dictionary approaches could not recognize the sentiments correctly in complex and incorrect-grammar feedbacks. For example, the feedback “this product does what it is supposed to do but I am not real happy with the quality of construction.” is a negative one. This feedback uses a complex sentence with incorrect grammar in the place of “not real happy”. The SentiStrength algorithm could not detect the words “not real” and it detected only one word “happy”. Therefore, its result was positive. However, our algorithm could detect all of three words “not real happy” and recognized it as a negative feedback because the algorithm considers on multiple levels and each sentiment word.

Furthermore, the complexity can be shown in how each algorithm handles the following feedback, “This is cheap and it feels cheap but it works and thats all that matters. I have one on the floor and two on the ceiling for months.” Human would interpret this example feedback as ‘Positive.’ However, the SentiStrength and Word-count algorithms reported it as ‘Negative’ and only the proposed algorithm revealed it as ‘Positive.’ The demonstration of detailed processes from each algorithm are shown in Table 4.1.

Table 4.1. The example processes of the three algorithms.

Algorithm	**Process**
Word-count	This is ***cheap*** **[Negative]** and it feels ***cheap*** **[Negative]** but it **works [Positive]** and thats all that matters. I have one on the floor and two on the ceiling for months.
	Positive count = 1 and Negative count = 2
	Positive < Negative = **“Negative”**
SentiStrength	-This is **cheap[-2]** and it feels **cheap[-2]** but it works and thats all that matters **.[sentence: 1,-2]** -I have one on the floor and two on the ceiling for months . **[sentence: 1,-1]**
	[result: max + and - of any sentence][overall result = -1 as pos<-neg] **“Negative”**
Our Algorithm	-This is **_cheap_[Neg1] [Negative]** **-**and it feels **_cheap_[Neg1] [Negative]** **-but[Contrast]** it **works [Pos1] [Contrast, Positive]** -and thats all that matters. -I have one on the floor -and two on the ceiling for months.
	[Negative**, Negative, Contrast, Positive**] = [Negative, **Positive**]
	Positive average = 50% **“Positive”**

Herein, the effectiveness of the algorithms were tested on two sentiment polarity labelled big datasets, Amazon and Yelp in [55]. The Yelp Review Polarity dataset consists of 560,000 positive and negative customer feedbacks and the Amazon Review Polarity dataset consists of 3,600,000 positive and negative feedbacks. The confusion matrix is a well-known measurement to reveal the accuracy of the information extraction algorithms. It consists of four parameters which are true positive (TP), false positive (FP), true negative (TN), and false negative (FN) as represented in Table 4.2.

Table 4.2. The confusion matrix table.

	Extracted: True	**Extracted: False**
Actual: True	True Positive (TP)	False Positive (FP)
Actual: False	False Negative (FN)	True Negative (TN)

Those variables are used for evaluation by applying the following performance measurements:

Precision = TP / (TP+FP)

Recall = TP / (TP+FN)

F-measure = 2*(Precision*Recall)/(Precision+Recall)

Accuracy = (TP+TN) / (TP+FP+FN+TN)

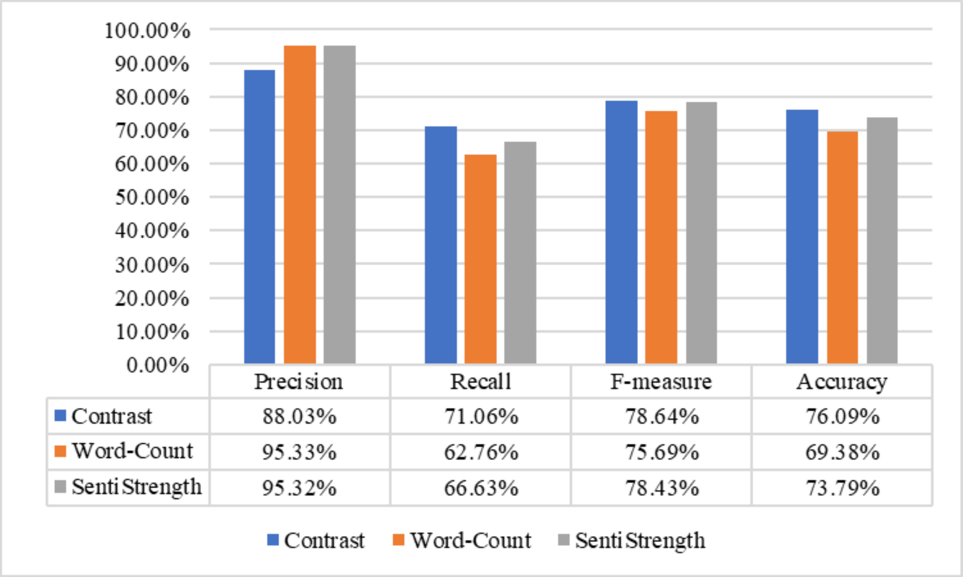
The confusion matrix results of the three algorithms on the Yelp Review Polarity dataset is shown in Table 4.3. The performance comparison is shown in Figure 4.1. Likewise, Table 4.4 and Figure 4.2 show the confusion matrix results and the performance comparison of the three algorithms on the Amazon Review Polarity dataset.

Table 4.3. Comparing the confusion matrix results on the Yelp Review Polarity dataset.

Matrix Variable	**Contrast**	**SentiStrength**	**Word-count**
True Positive (TP)	246,474 (88.03%)	266,898 (95.32%)	266,932 (95.33%)
False Positive (FP)	33,526 (11.97%)	13,102 (4.68%)	13,068 (4.67%)
True Negative (TN)	179,641 (64.16%)	146,329 (52.26%)	121,621 (43.44%)
False Negative (FN)	100,359 (35.84%)	133,671 (47.74%)	158,379 (56.56%)

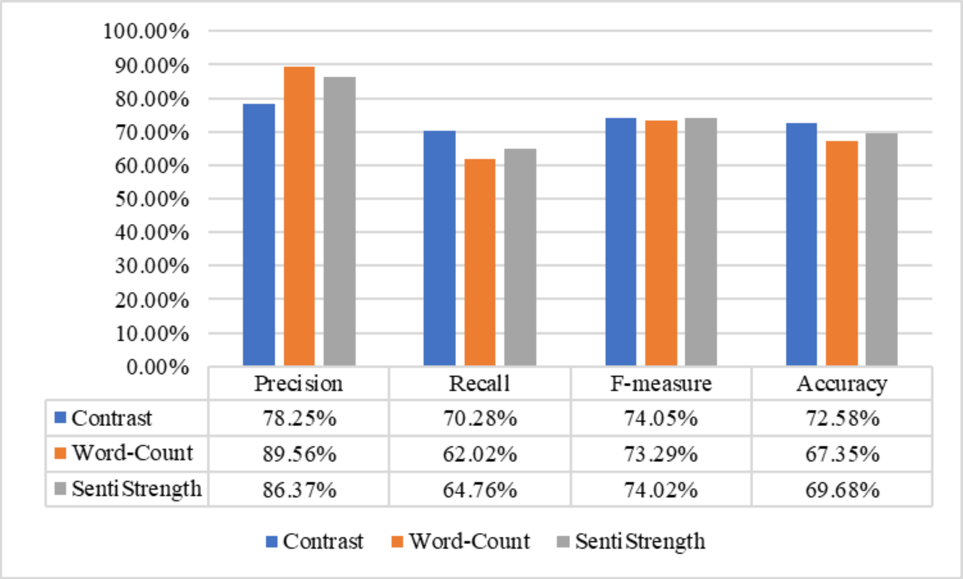
Table 4.4. Comparing the confusion matrix results on the Amazon Review Polarity dataset.

Matrix Variable	**Contrast**	**SentiStrength**	**Word-count**
True Positive (TP)	1,408,548 (78.25%)	15,554,601 (86.37%)	1,612,135 (89.56%)
False Positive (FP)	391,452 (21.75%)	245,399 (13.63%)	187,865 (10.44%)
True Negative (TN)	1,204,353 (66.91%)	953,912 (53.00%)	812,552 (45.14%)
False Negative (FN)	595,647 (33.09%)	846,088 (47.00%)	987,448 (54.86%)



	Precision	Recall	F-measure	Accuracy
Contrast	88.03%	71.06%	78.64%	76.09%
Word-Count	95.33%	62.76%	75.69%	69.38%
SentiStrength	95.32%	66.63%	78.43%	73.79%

Figure 4.1. The performance measurement on the Yelp Review Polarity dataset.



	Precision	Recall	F-measure	Accuracy
Contrast	78.25%	70.28%	74.05%	72.58%
Word-Count	89.56%	62.02%	73.29%	67.35%
SentiStrength	86.37%	64.76%	74.02%	69.68%

Figure 4.2. The performance measurement on the Amazon Review Polarity dataset.

Tables 4.3 and 4.4 show the confusion matrix results of each algorithm on each dataset. The Word-count algorithm achieved slightly more correct results than others in the “positive” polarity on both datasets. However, the proposed contrast algorithm could recognize the “negative” polarity more correctly than the other algorithms. The Contrast Dictionary algorithm recognized more true “negative” transactions, 179,641 from 280,000 “negative” transactions (64.16%) of the Yelp Review Polarity dataset, comparing to 146,329 or 52.26% of the SentiStrength and

121,621 or 43.44% of the Word-count. Likewise, the proposed algorithm recognized 1,204,353 out of 1,800,000 negative transactions (66.91%) of the Amazon Review Polarity dataset. It is common that there are more negative feedbacks than positive ones. The proposed algorithm's prediction was more correct than the others. The results on negative transactions added to the total accuracy help the proposed algorithm to perform the best in the comparison.

Figures 4.1 and 4.2 represent the comparison of the performance measurements of the three algorithms on each dataset. The contrast algorithm performed much better than the others in these two measurements. Contrarily, the 'Precision' measures the recognition rate of the "positive" polarity. The proposed algorithm performed the worst in the 'Precision' measurement on both datasets. However, the proposed algorithm attained in the performance measurement of the 'Recall' which measures the sensitivity of an algorithm on a dataset. The contrast algorithm accomplished slightly better than the others in the 'F-measure' performance that measures the test's accuracy of an algorithm. Therefore, the proposed algorithm achieved the highest accuracy in the 'Accuracy' measurement. It is because the proposed algorithm considers more possible sentiment features from customer feedbacks on multiple levels: word, sentence, contrast, and document. Considering multiple levels of sentiment analysis contributes to a significant difference to the others. The Word-count algorithm considers only on the counts of positive and negative sentiment words and SentiStrength recognizes the sentiment information based on the level values of positive and negative words.

4.2 The Aspect-Based Sentiment Information Extraction

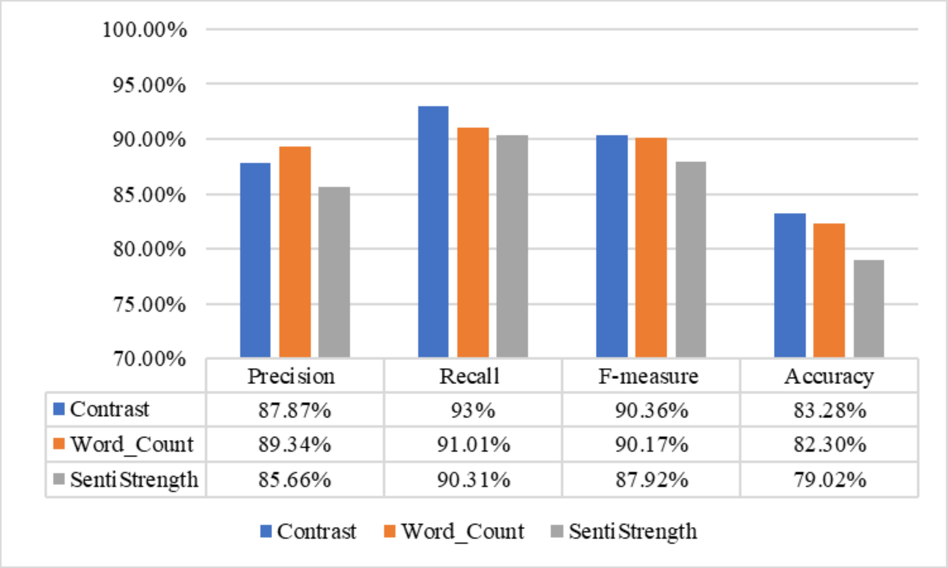
Even though our Contrast Dictionary algorithm achieved the highest accuracy on the both of big datasets, it is not enough to apply in real-world applications. For this reason, a hybrid approach is proposed, called the aspect-based sentiment information extraction. It aims to improve the sentiment performances. Its effectiveness was tedted on the two small datasets (training and testing dataset). The training dataset for aspects consists of 305 customers' feedbacks selected from the Amazon's Automotive dataset based on product IDs [53], [54]. It consists of 272 "Positive" and

33 “Negative” feedbacks, and, of the total 305 feedbacks, 227 feedbacks include aspects. The testing dataset include other 100 customers’ feedbacks on the five Amazon’s Automotive products. It consists of 50 “Positive” and “Negative” feedbacks.

The confusion matrix comparing the three algorithms on the training dataset (305 transactions) without checking the aspects is shown in Table 4.5. The Word-count algorithm slightly got more correct results than the others in the “positive” polarity. However, the contrast algorithm could recognize the “negative” polarity more correctly than the other algorithms. The performance comparison of the three algorithms is shown Figure 4.3.

Table 4.5. The confusion matrix results on the training dataset without checking on the aspects.

Matrix Variable	Contrast	Word-count	SentiStrength
True Positive (TP)	239 (87.87%)	243 (89.34%)	233 (85.66%)
False Positive (FP)	33 (12.13%)	29 (10.66%)	39 (14.34%)
True Negative (TN)	15 (45.45%)	9 (27.27%)	8 (24.24%)
False Negative (FN)	18 (54.55%)	24 (72.73%)	25 (75.75%)



	Precision	Recall	F-measure	Accuracy
■ Contrast	87.87%	93%	90.36%	83.28%
■ Word_Count	89.34%	91.01%	90.17%	82.30%
■ Senti Strength	85.66%	90.31%	87.92%	79.02%

Figure 4.3. The comparison of sentiment extraction on the three algorithms on the training dataset.

Figure 4.3 shows the performance measurements of the three algorithms on the 305 records of training dataset. The proposed contrast algorithm gets the highest accuracy than the others. However, the aspect-based sentiment can further enhance the extracted sentiment information of these three algorithms. The matrix values on the training dataset checking with the aspect-based is shown in Table 4.6. The measurement of aspect-based sentiment extraction on the dataset is shown in Figure 4.4.

Table 4.6. The confusion matrix results on the training dataset with aspect sentiment analysis.

Matrix Variable	**Contrast with Aspect**	**Word-count with Aspect**	**SentiStrength with Aspect**
True Positive (TP)	247 (90.81%)	256 (94.12%)	247 (90.91%)
False Positive (FP)	25 (9.19%)	16 (5.88%)	25 (9.19%)
True Negative (TN)	21 (63.64%)	16 (48.48%)	15 (45.45%)
False Negative (FN)	12 (36.36%)	17 (51.52%)	18 (54.55%)

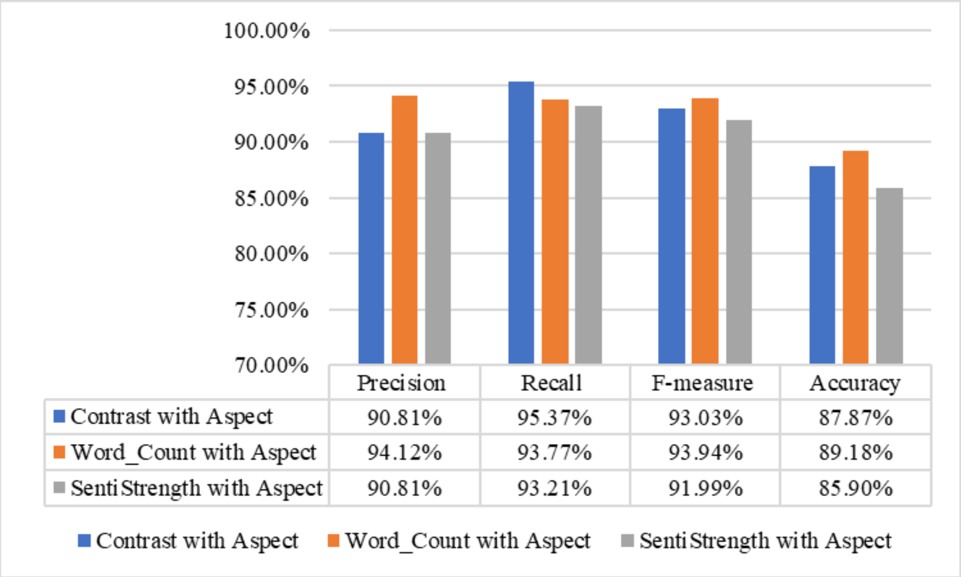


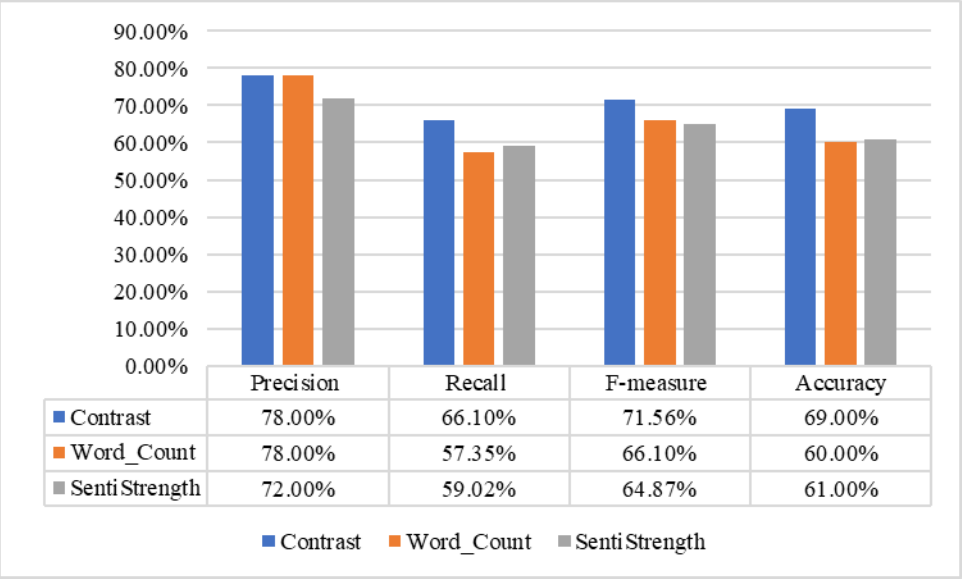
Figure 4.4. The comparison of aspect-based enhance sentiment extraction on the training dataset.

Figure 4.4 shows the performance comparison on the training dataset of three algorithms enhanced with the aspect-level sentiment analysis. The Word-count algorithm with aspect analysis gets the highest accuracy than the others. The contrast algorithm with aspect analysis reaches moderate accuracy and the highest in recall. It means the contrast algorithm can detect the “negative” sentiment more correctly than the others. However, we can see the improvement of each algorithm in each measurement.

The results on the training dataset with aspect analysis obtained impressive accuracy. Table 4.7 shows the results on the 100 records of the testing dataset. The proposed algorithm achieved significantly more correct results than the others in the both “positive” and “negative” polarities. The performance comparison of the three algorithms without aspect analysis on the testing dataset is shown in Figure 4.5.

Table 4.7. The confusion matrix results on the testing dataset without the aspect analysis.

Matrix Variable	**Contrast**	**Word-count**	**SentiStrength**
True Positive (TP)	39 (78.00%)	39 (78.00%)	36 (72.00%)
False Positive (FP)	11 (22.00%)	11 (22.00%)	14 (28.00%)
True Negative (TN)	30 (60.00%)	21 (42.00%)	25 (50.00%)
False Negative (FN)	20 (40.00%)	29 (58.00%)	25 (50.00%)



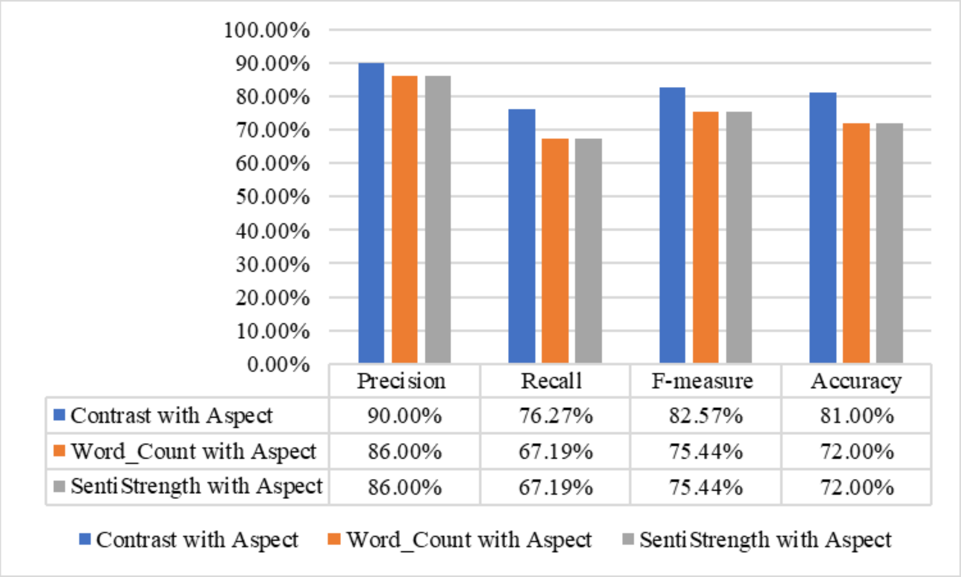
	Precision	Recall	F-measure	Accuracy
■ Contrast	78.00%	66.10%	71.56%	69.00%
■ Word_Count	78.00%	57.35%	66.10%	60.00%
■ SentiStrength	72.00%	59.02%	64.87%	61.00%

Figure 4.5. The performance comparison of three sentiment extraction algorithms without aspect analysis on the testing dataset.

The proposed algorithm got the highest performance than the others. On one hand, the aspect-level sentiment extraction is applied using the Apache OpenNLP [74] combining the N-gram theory [76] to extract the sentiment information on each desired aspect. This approach correctly detects 144 aspect sentiments from the total 173 aspects on the testing dataset. The testing dataset includes 230 aspects mentioned in the 100 transactions. The model could detect 75.22% of the aspects. It can help improve the performance of sentiment extraction. The matrix results on the testing dataset with checking on aspects is shown in Table 4.8 and the performance measurement is shown in Figure 4.6.

Table 4.8. The confusion matrix results on the testing dataset with aspect analysis.

Matrix Variable	**Contrast with Aspect**	**Word-count with Aspect**	**SentiStrength with Aspect**
True Positive (TP)	45 (90.00%)	43 (86.00%)	43 (86.00%)
False Positive (FP)	5 (10.00%)	7 (14.00%)	7 (14.00%)
True Negative (TN)	36 (72.00%)	29 (58.00%)	29 (58.00%)
False Negative (FN)	14 (28.00%)	21 (42.00%)	21 (42.00%)



	Precision	Recall	F-measure	Accuracy
■ Contrast with Aspect	90.00%	76.27%	82.57%	81.00%
■ Word_Count with Aspect	86.00%	67.19%	75.44%	72.00%
■ SentiStrength with Aspect	86.00%	67.19%	75.44%	72.00%

Figure 4.6. The performance comparison of three sentiment extraction algorithms with aspect analysis on the testing dataset.

Figure 4.6 shows the performance measurement of aspect-based sentiment extraction on the 100-record testing dataset. Even though it does not reach as high accuracy as appeared in the training dataset, it has improved the overall accuracy of the three sentiment extraction algorithms. The proposed algorithm with aspect analysis got the highest accuracy than the others. Its accuracy has improved from 69% to 81%. The SentiStrength and the Word-count algorithms with aspect analysis also reached the same accuracy on the testing dataset. SentiStrength increased the accuracy from 61% to 72% and the Word-count from the 60% to the 72% respectively.

In summary, the contrast dictionary algorithm can extract the sentiment information more correctly than the SentiStrength and Word-count whether applying the aspect analysis or not. It is because the algorithm can handle complex sentences better than the others and more correctly recognize the ‘negative’ polarity than the others. The aspect-based sentiment information extraction can help improve the performance of sentiment extraction from customer feedbacks in all three algorithms.

4.3 Results (Production categorization applying the RFC model)

The objective of the RFC model is to determine the product status comparing with similar products based on the interested time frame. This means that the marketers can see the changes of products situations and make a decision. They can easily understand and respond to the situation changes by applying some marketing strategies to win the customers’ interests and satisfactions.

This is similar to the aim of the RFM model in customer segmentation technique in traditional market [4], [52], [56]. Even though the RFC model may allow the user to classify the customers and products in SMBD, this work focuses on only the products. The RFM are applied in many domains combining with data mining techniques [4]. Nonetheless, they never focus on social market classifying the products based on the customers’ feedbacks.

As there have been increasing interests in product information from SMBD, most of them extracted only one of the main three key variables (Recency, Frequency, and Credit) information. They did not combine of these variables to represent deeper product information. The feature comparison of previous works and our product categorization approach is shown in Table 4.9. This work aims to extract and employ all three variables for a marketing recommendation system.

Table 4.9. Feature comparison of product information extraction approaches.

Literature	Approach	Credit	Frequency	Recency
(Li et al., 2017), (Smailović, Grčar, Lavrač, & Žnidaršič, 2013), (Asur & Huberman, 2010), (Gruhl, Guha, Kumar, Novak, & Tomkins, 2005)	Predicting the stock markets based on the sentiment or credit from SMBD	✓		
(Alengadan & Khan, 2017), (Gardner & Scott, 1990)	Ranking the products based on the sentiment frequencies	✓	✓	
(Kendall, 2018)	Extracting movement information of financial markets based on the time cost			✓
(Castro & Tsuzuki, 2015), (Mohammadian & Makhani, 2016), (Cheng & Chen, 2009)	Segmenting the customers using the RFM model		✓	✓
Our Approach	Categorizing the products using the RFC model	✓	✓	✓

The product categorization helps reveal the extracted RFC information of products from SMBD. The users can notice the best products which have recent and numerous feedbacks with high credit and the poor products which are low in recency, frequency, and credit, in certain period. The effectiveness of the RFC model mainly depends on the accuracy of the C value from feedback sentiments. Applying the RFC model on customer feedbacks from ten Amazon’s Automotive products including 305 transactions in five separated quarter periods, each quarter involves three months of a year, and the last quarter is the current period. The extracted information of the R, F, and C values are shown in Figures 4.7, 4.8, and 4.9.

The R value is the difference between the last feedback date and the beginning of the period, the F value shows the total number of transactions, and the C value represents the accumulation of feedbacks’ sentiments on a product in each period. Herein, we can see the moving information of the R, F, and C quarterly. In Figure 4.7, the R value represents the activeness of a product in a certain period. The smaller number is better on the R variable. Although the product “**P1**” with the blue line was the most in-recently active (R) in the first quarter of the previous year, it moved up and become the most recent active in the last quarter. This means that the product recently had customer interests.

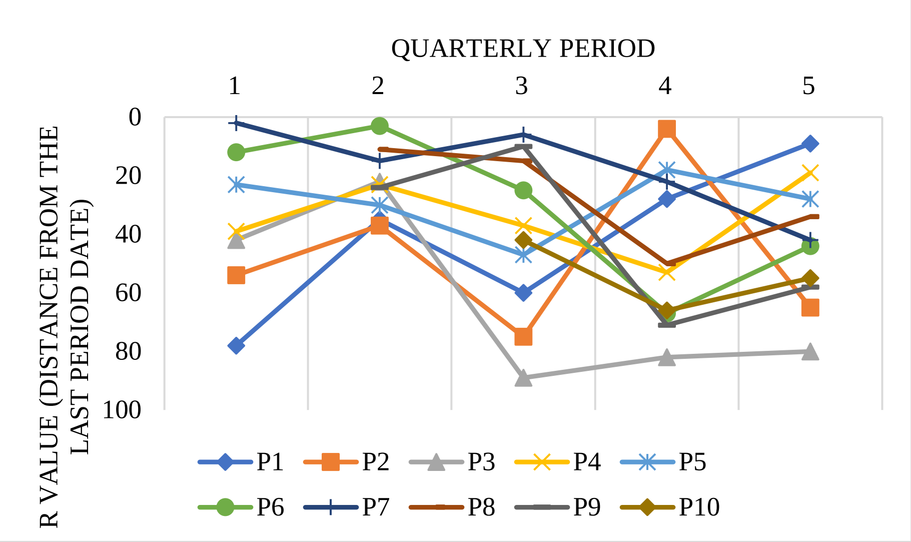


Figure 4.7. Current R values in each quarter.

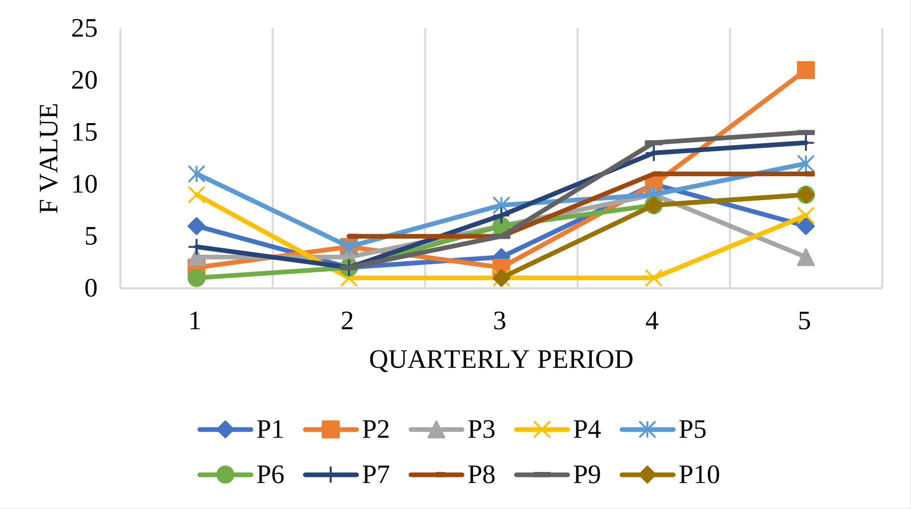


Figure 4.8. Current F values in each quarter.

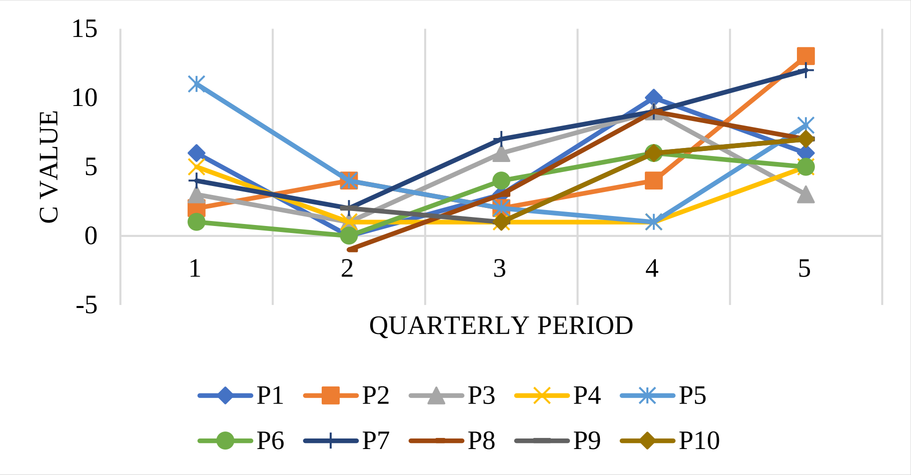


Figure 4.9. Current C values in each quarter.

The F value shows the number of customer interests on a product and the C value describes the accumulation of customers' satisfactions in a period. Comparing the F and C values, the marketers can get more insights about their products. For instance, the product "**P8**" with the brown line started to active in the second quarter and had the highest F number, but the C value is very low. This means that a lot of customers were unsatisfied about this product.

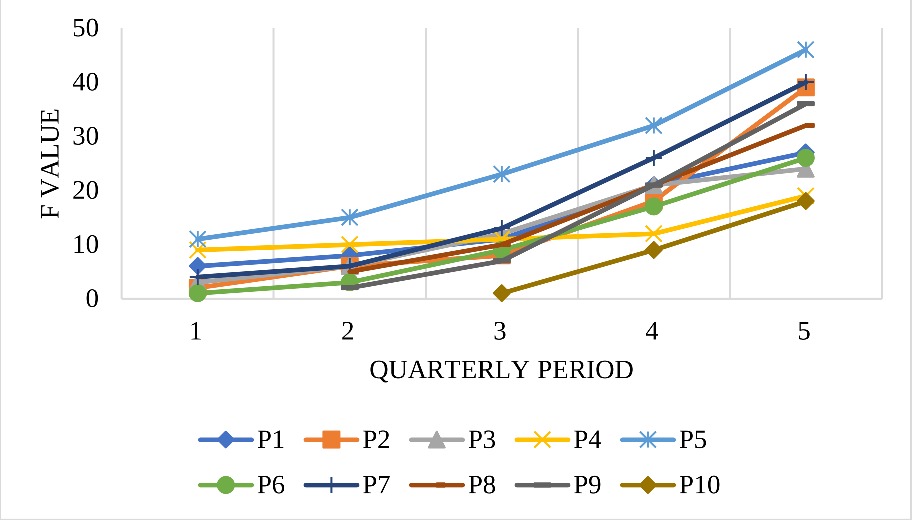


Figure 4.10. Accumulated F values in each quarter.

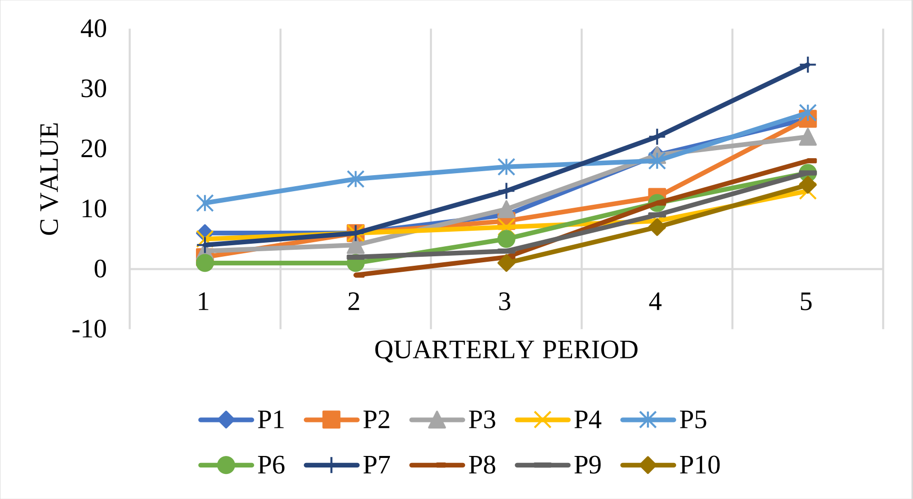


Figure 4.11. Accumulated C values in each quarter.

The accumulated F and C values in each quarter and the RFC model product categorization are represented in Figures 4.10 and 4.11. The numbers from 1 to 5 of the x-axes describe the five quarter periods and the numbers from "**P1**" to "**P10**" represent ten different products. Herein, the accumulation of the F value always increases in each period, but the accumulation of the C value can be fluctuated. However, the marketers can see the improvement of the product period by period.

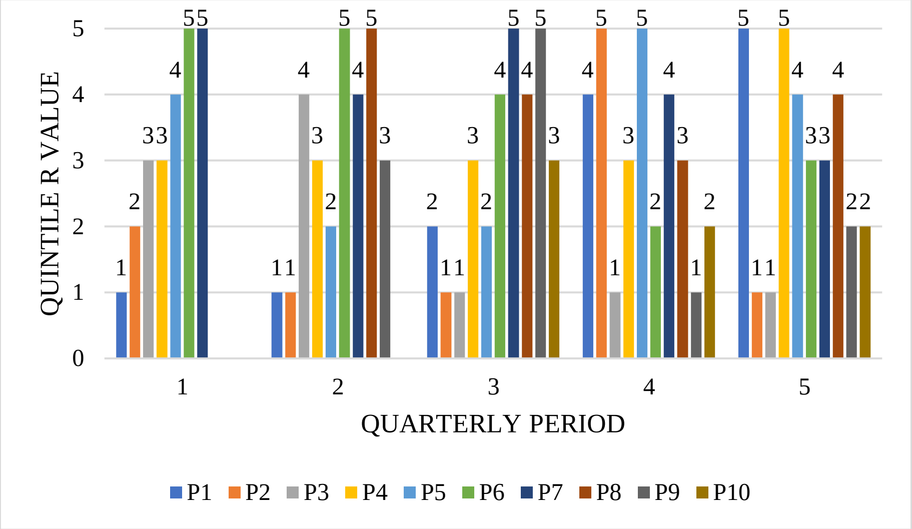


Figure 4.12. Quintile information of the R values in each quarter.

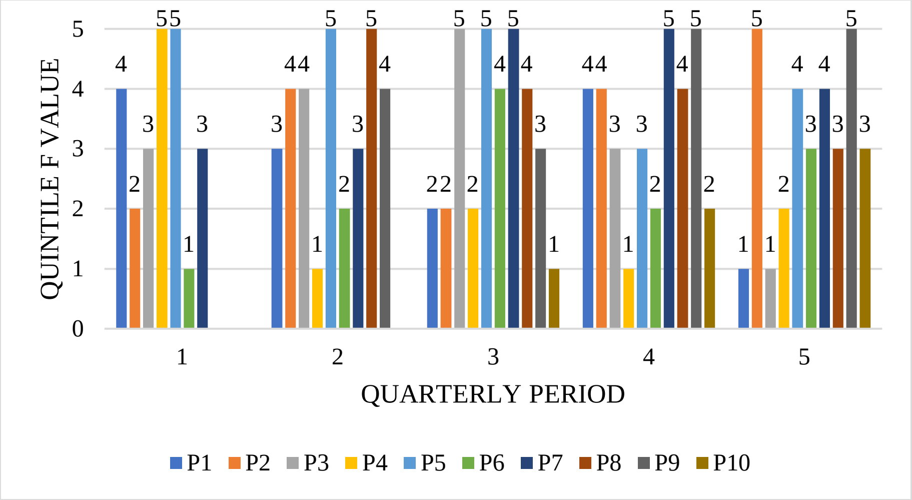


Figure 4.13. Quintile information of the F values in each quarter.

The quintile analysis divides all population into five different groups. These three RFC values are information of products which can help in categorizing the products. The data changes all the time. The quintile algorithm is the most suitable technique to report such dynamic data. It divides the information of the R, F, and C values into five different sets as shown in Figures 4.12, 4.13, and 4.14. The smallest number is 1 and the highest number is 5. The product movements are shown in each period. Moreover, these five sets of variables are also applied in the rule patterns with the RFC model for product categorization. The extracted eight product categories on the example of ten products is as shown in Figure 4.15.

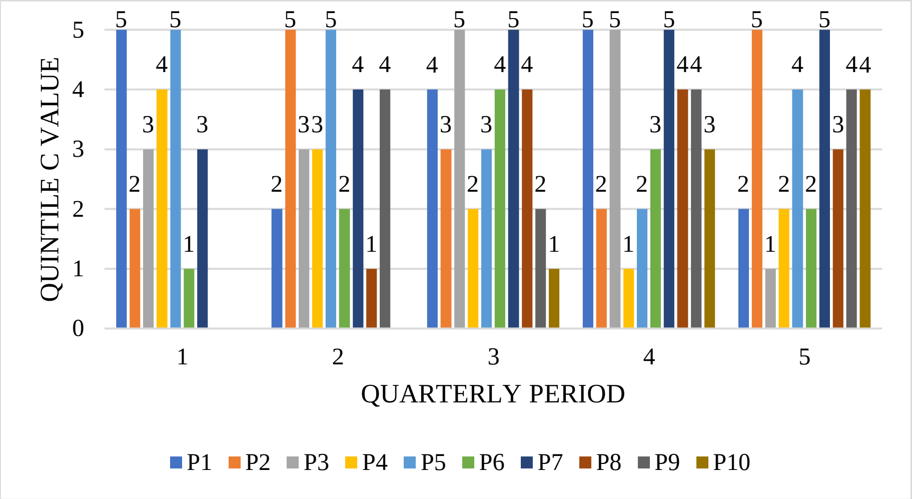


Figure 4.14. Quintile information of the C values in each quarter.

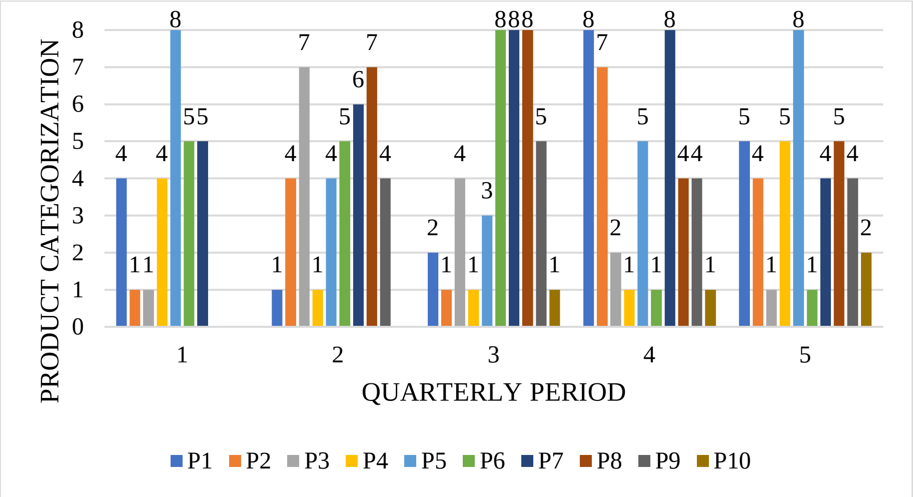


Figure 4.15. The RFC model product categorization in each quarter.

The product categorization is based on the quintile analysis and rule-based algorithms. The numbers from 1 to 8 in Figure 4.15 represent the eight different product categories or clusters from the RFC patterns in the Table 3.11. The number 1 is the “**Poorest Product**” which has all down values of the RFC (R↓F↓C↓) and 8 is the “**Best Product**” that has all up variables (R↑F↑C↑). The marketers can recognize customer satisfactions on their products based on these eight product categories. For example, the product “**P2**” with the orange line with orange line was in “**category 1**” (Poorest Product) in the first quarter which means the product had a few customers with negative opinions. However, it significantly rose to the "**category 4**" in the second quarter. This means that recently the product did not have customers interests, but it had high frequency and credit during this period. In the third quarter, it was going down again into the "**category 1**." However, it became "**category 7**" in the fourth quarter which means that the product had recently customers' interests with a high frequency, but most of them were negative. The marketer should respond to the negative customers and improve their product market. It was going to "**category 4**" in the last period.

The alarming system can help the marketers notify the changes of products information based on the rules. It will notice the marketers on the R, F, and C information and the product category changes. For example, the product "**P6**" with the green line had the highest recency in the second quarter and it was significantly dropped down in the third and fourth quarters. At that time the system should notify the marketers that the product has no customer interests or feedbacks in this period. They should apply a suitable marketing strategy to sustain their products stability in the current market. The system should notify not only on recency changes, but also every information changes.

4.4 Summary

This chapter validated and verified the proposed approaches, Contrast Dictionary for sentiment extraction, Aspect-based sentiment to improve sentiment information extraction accuracy, and the RFC model of product information extraction. The effectiveness of the Contrast Dictionary tested on two big datasets and the aspect-based sentiment tested on two small datasets. The proposed approaches got the highest accuracies on every testing. Moreover, this compared the researches on SMBD related with the R, F, and C variables, and discussed an example of the RFC model applying on a social market analytics system.

CHAPTER 5
CONCLUSIONS

The chapter involves five sections related to overall conclusions of the research. Section 5.1 presents the summary of the thesis. The contributions of this thesis are described in the Section 5.2. Section 5.3 presents the limitations and shortcomings of the thesis works are shown in the Section 5.4. Finally, the future works are shown in the Section 5.4.

5.1 Summary

In this information age, the growing up of the data is difficult to extract a new knowledge to support the information management, decision-making, and recommendation systems. In terms of social marketing, many researches and industries have been tried to extract the useful information from SMBD, applying several approaches, methods and algorithms in order to support the marketers to make better decisions. The tracking or monitoring the customer sentiments is play an important row in SMBD analysis. Herein, the proposed contrast dictionary algorithm can handle the complexity of human languages more than the standard algorithms and provides more accurate results from feedbacks, especially in the negative feedbacks. Moreover, the hybrid sentiment extraction approach (aspect-based sentiment extraction) provides more accurate sentiment results of customer feedbacks on the SMBD. The marketers can understand their customers’ satisfactions from the feedbacks in deep such as not only the sentiment and also the sentiments on each aspect of a product.

On the other hand, the RFM model is a flavor technology of information extraction to support the marketing strategies, especially for the traditional marketing. However, they could not support to apply in the SMBD in order to support the social marketing. To solve this problem, the thesis researched on the RFM model and information extraction, especially in the social market analysis. Then, we proposed an adaptative model of the RFM which called the RFC model is proposed for the SMBD analytics.

In the proposed RFC model, R means the recent date of customer feedbacks on the product, F means the total numbers of feedbacks occurring in the interesting period, and the C comes from the customer sentiments extracted by the two proposed sentiment algorithms: contrast dictionary and aspect-based algorithm.

Due to the unstructured forms of the SMBD, the RFC required the advanced technologies (data science techniques) to deal with these forms. Herein, the NLP and OM are the key technologies to transform the unstructured data into the structured forms in terms of information extraction. The DM is a technology to discover new knowledges from the structured data that the outputs of the NLP and OM. This knowledge supported the marketers to consider in their marketing strategies such as how to create a strategy to get the customers' interested.

5.2 Claims to Originality

The research mainly contributes the information extraction approaches from the SMBD in order to support the decision making and recommendation systems for social markets. This proposed three main contributions with different three objectives as shown below:

1. The contrast dictionary algorithm aimed to handle the complexity of human languages from the SMBD. This could control the SMBD better than the two well-known approaches of the SentiStrength and Word-count algorithms.

2. The aspect-based sentiment information extraction approach desired to improve the performance of the sentiment extraction from the SMBD. The experimental results showed that our approach could significantly improve the sentiment information extraction performances.

3. The RFC model was a main contribution of the thesis which purposed to apply in the SMBD analytics system to produce the product information in order to support the social marketing. This research provided the R, F, and C variables information on each product and the products categories based on that information.

5.3 Limitations

The contrast dictionary algorithm is mainly depended on the dictionary of collected corpus features of “Positive,” “Negative,” “Negation,” and “Contrast” words. The system uses over 2000 “Positive words” and over 4000 “Negative words” collected by the Professor Bing Liu [9], [79] combining with the library of the SentiStrength [61]. The “Negation” and “Contrast” are used well-known words, for instance, the “Negation” words are ‘no’ and ‘not’ and the “Contrast” words are ‘but’ and ‘although.’ The aspect-based sentiment approach specially relies on the aspect features from feedbacks on products because different products have different features and aspects. The scopes of RFC model are similar to the RFM model. However, to apply the RFC model on a SMBD analytics mainly depends on the correctness of the sentiment or credit (C) information extraction.

5.4 Shortcomings

The effectiveness of the contrast dictionary algorithm has been tested on the two labelled sentiment datasets. It got the highest accuracies on both datasets. However, the aspect-based sentiment information extraction and the RFC model were not tested on big datasets due to the time limitations. These two approaches have been tested on small customer feedbacks datasets from the Amazon’s Automotive products.

5.5 Future Works

This thesis proposes the concept of the RFC model and sentiment extractions. In reality, the RFC can be applied in many areas in the same ways with the RFM model. The RFC can be applied in the SMBD analysis, serving several purposes as shown below:

- Prediction the future product situations based on the fluctuation of moving R, F, and C variables information
- Customer behaviors segmentation based on the customer feedbacks on the SMBD
- Extracting the target customer for products’ promotions

- Recommendations or suggestions on products based on the demands and supplies in vital marketing.

In further research, the sentiment information extraction from customer feedbacks can be improved to build a hybrid model of Bag-of-words, machine learning, and deep learning.

BIBLOGRAPHY

[1] M. Mohammadian and I. Makhani, “RFM-based customer segmentation as an elaborative analytical tool for enriching the creation of sales and trade marketing strategies,” *Int. Acad. J. Account. Financ. Manag.*, vol. 3, no. 6, pp. 21–35, 2016.

[2] L. Pépin, P. Kuntz, J. Blanchard, F. Guillet, and P. Suignard, “Visual analytics for exploring topic long-term evolution and detecting weak signals in company targeted tweets,” *Comput. Ind. Eng.*, vol. 112, pp. 450–458, Oct. 2017.

[3] Y. S. Patel, D. Agrawal, and L. S. Josyula, “The RFM-based ubiquitous framework for secure and efficient banking,” in *2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, pp. 283–288, 2016.

[4] D. Birant, “Data Mining Using RFM Analysis,” *Knowledge-Oriented Appl. Data Min.*, pp. 91–108, 2011.

[5] J. T. Wei, S. Y. Lin, Y. Z. Yang, and H. H. Wu, “Applying Data Mining and RFM Model to Analyze Customers’ Values of a Veterinary Hospital,” in *2016 International Symposium on Computer, Consumer and Control (IS3C)*, pp. 481–484, 2016.

[6] E. G. Castro and M. S. G. Tsuzuki, “Churn Prediction in Online Games Using Players’ Login Records: A Frequency Analysis Approach,” *IEEE Trans. Comput. Intell. AI Games*, vol. 7, no. 3, pp. 255–265, Sep. 2015.

[7] C.-H. Cheng and Y.-S. Chen, “Classifying the segmentation of customer value via RFM model and RS theory,” *Expert Syst. Appl.*, vol. 36, no. 3, Part 1, pp. 4176–4184, Apr. 2009.

[8] E. Wong and Y. Wei, “Customer online shopping experience data analytics,” *Int. J. Retail Distrib. Manag.*, vol. 46, no. 4, pp. 406–420, Apr. 2018.

[9] M. Hu and B. Liu, “Mining and summarizing customer reviews,” in *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’04*, p. 168, 2004.

[10] C. Diamantini, A. Mircoli, D. Potena, and E. Storti, “Social information discovery enhanced by sentiment analysis techniques,” *Futur. Gener. Comput. Syst.*, Feb. 2018.

[11] D. T. Nguyen and J. E. Jung, “Real-time event detection for online behavioral analysis of big social data,” *Futur. Gener. Comput. Syst.*, vol. 66, pp. 137–145, Jan. 2017.

[12] A. Olteanu, O. Varol, and E. Kiciman, “Distilling the Outcomes of Personal Experiences,” in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing - CSCW ’17*, pp. 370–386, 2017.

[13] “• Twitter by the Numbers (2019): Stats, Demographics &amp; Fun Facts.” [Online]. Available: https://www.omnicoreagency.com/twitter-statistics/. [Accessed: 23-May-2019].

[14] Y.-L. Hwong, C. Oliver, M. Van Kranendonk, C. Sammut, and Y. Seroussi, “What makes you tick? The psychology of social media engagement in space science communication,” *Comput. Human Behav.*, vol. 68, pp. 480–492, Mar. 2017.

[15] L. I. Labrecque and K. Swani, “Does Linguistic Style in Social Media Communications Impact Consumer Engagement? An Abstract,” in *Creating Marketing Magic and Innovative Future Marketing Trends*, Springer, Cham, pp. 597–598, 2017.

[16] B. Li, K. C. C. Chan, C. Ou, and S. Ruifeng, “Discovering public sentiment in social media for predicting stock movement of publicly listed companies,” *Inf. Syst.*, vol. 69, pp. 81–92, Sep. 2017.

[17] A. Rossmann, T. Wilke, and G. Stei, “Usage of Social Media Systems in Customer Service Strategies,” in *Hawaii International Conference on System Sciences*, pp. 3950–3959, 2017.

[18] C. Oh, Y. Roumani, J. K. Nwankpa, and H.-F. Hu, “Beyond likes and tweets: Consumer engagement behavior and movie box office in social media,” *Inf.*

Manag., vol. 54, no. 1, pp. 25–37, Jan. 2017.

[19] W. Fan and M. D. Gordon, “The power of social media analytics,” *Commun. ACM*, vol. 57, no. 6, pp. 74–81, Jun. 2014.

[20] S. Teng, K. W. Khong, A. Y. L. Chong, and B. Lin, “Persuasive electronic word-of-mouth messages in social media,” *J. Comput. Inf. Syst.*, vol. 57, no. 1, pp. 76–88, Jan. 2017.

[21] G. Bello-Orgaz, J. J. Jung, and D. Camacho, “Social big data: Recent achievements and new challenges,” *Inf. Fusion*, vol. 28, pp. 45–59, Mar. 2016.

[22] U. Sivarajah, M. M. Kamal, Z. Irani, and V. Weerakkody, “Critical analysis of Big Data challenges and analytical methods,” *J. Bus. Res.*, vol. 70, pp. 263–286, Jan. 2017.

[23] J. J. Jung, “Computational Collective Intelligence with Big Data: Challenges and Opportunities,” *Futur. Gener. Comput. Syst.*, vol. 66, pp. 87–88, Jan. 2017.

[24] M. Safiullah, P. Pathak, S. Singh, and A. Anshul, “Social media as an upcoming tool for political marketing effectiveness,” *Asia Pacific Manag. Rev.*, vol. 22, no. 1, pp. 10–15, Mar. 2017.

[25] T. Hopp and C. J. Vargo, “Does negative campaign advertising stimulate uncivil communication on social media? Measuring audience response using big data,” *Comput. Human Behav.*, vol. 68, pp. 368–377, Mar. 2017.

[26] T. Liu, Y. Zhong, and K. Chen, “Interdisciplinary study on popularity prediction of social classified hot online events in China,” *Telemat. Informatics*, vol. 34, no. 3, pp. 755–764, Jun. 2017.

[27] H. Baek, S. Oh, H.-D. Yang, and J. Ahn, “Electronic word-of-mouth, box office revenue and social media,” *Electron. Commer. Res. Appl.*, vol. 22, pp. 13–23, Mar. 2017.

[28] J.-J. Sheu and K.-T. Chu, “Mining association rules between positive word-of-mouth on social network sites and consumer acceptance: A study for derivative product of animations, comics, and games,” *Telemat. Informatics*, vol. 34, no. 4,

pp. 22–33, Jul. 2017.

[29] S. S. C. Shang, Y.-L. Wu, and Y.-J. Sie, “Generating consumer resonance for purchase intention on social network sites,” *Comput. Human Behav.*, vol. 69, pp. 18–28, Apr. 2017.

[30] J. Lin, R. Oentaryo, E.-P. Lim, C. Vu, A. Vu, and A. Kwee, “Where is the Goldmine?: Finding Promising Business Locations Through Facebook Data Analytics,” in *Proceedings of the 27th ACM Conference on Hypertext and Social Media*, pp. 93–102, 2016.

[31] J. Smailović, M. Grčar, N. Lavrač, and M. Žnidaršič, “Predictive Sentiment Analysis of Tweets: A Stock Market Application,” Springer, Berlin, Heidelberg, pp. 77–88, 2013.

[32] S. Asur and B. A. Huberman, “Predicting the Future with Social Media,” in *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pp. 492–499, 2010.

[33] S. Tartir and I. Abdul-Nabi, “Semantic Sentiment Analysis in Arabic Social Media,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 29, no. 2, pp. 229–233, Apr. 2017.

[34] L. I. Tan, W. S. Phang, K. O. Chin, and P. Anthony, “Rule-Based Sentiment Analysis for Financial News,” in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1601–1606, 2015.

[35] D. Vilares, C. Gómez-Rodríguez, and M. A. Alonso, “Universal, unsupervised (rule-based), uncovered sentiment analysis,” *Knowledge-Based Syst.*, vol. 118, pp. 45–55, Feb. 2017.

[36] M. Zaw and P. Tandayya, “Multi-level Sentiment Information Extraction Using the CRbSA Algorithm,” in *2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 1–6, 2018.

[37] Z. Piao, S.-M. Park, B.-W. On, G. Choi, and M.-S. Park, “Product reputation mining: bring informative review summaries to producers and consumers,”

Comput. Sci. Inf. Syst., vol. 16, no. 00, pp. 6–6, 2019.

[38] E. Cambria, “Affective Computing and Sentiment Analysis,” *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar. 2016.

[39] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, “Enhancing deep learning sentiment analysis with ensemble techniques in social applications,” *Expert Syst. Appl.*, vol. 77, pp. 236–246, Jul. 2017.

[40] T. Chen, R. Xu, Y. He, and X. Wang, “Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN,” *Expert Syst. Appl.*, vol. 72, pp. 221–230, Apr. 2017.

[41] X. Fang and J. Zhan, “Sentiment analysis using product review data,” *J. Big Data*, vol. 2, no. 1, p. 5, Dec. 2015.

[42] B. B. Alengadan and S. S. Khan, “A proposed system for modifying aspect based opinion mining for ranking of products,” in *2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS)*, pp. 335–338, 2017.

[43] N. Majumder, S. Poria, A. Gelbukh, M. S. Akhtar, E. Cambria, and A. Ekbal, “IARM: Inter-Aspect Relation Modeling with Memory Networks in Aspect-Based Sentiment Analysis,” pp. 3402–3411, 2019.

[44] A. R. Alaei, S. Becken, and B. Stantic, “Sentiment Analysis in Tourism: Capitalizing on Big Data,” *J. Travel Res.*, vol. 58, no. 2, pp. 175–191, Feb. 2019.

[45] S. Du, J. Wang, and K. Gwebu, “Stock market reaction to data breaches: The moderating role of corporate social responsibility,” in *2017 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA)*, pp. 1–2, 2017.

[46] M. P. Gardner and J. Scott, “Product Type: a Neglected Moderator of the Effects of Mood,” *ACR North Am. Adv.*, vol. 17, pp. 585–589, 1990.

[47] C. Kendall, “The time cost of information in financial markets,” *J. Econ. Theory*, vol. 176, pp. 118–157, Jul. 2018.

[48] MISHNE and G., “Pridicting movie sales from blogger sentiment,” *AAAI 2006*

Spring Symp. Comput. Approaches to Anal. Weblogs, pp. 155–158, 2006.

[49] S. Sun, C. Luo, and J. Chen, “A review of natural language processing techniques for opinion mining systems,” *Inf. Fusion*, vol. 36, pp. 10–25, Jul. 2017.

[50] G. G. Chowdhury, “Natural language processing,” *Annu. Rev. Inf. Sci. Technol.*, vol. 37, no. 1, pp. 51–89, Jan. 2005.

[51] C. Catal and M. Nangir, “A sentiment classification model based on multiple classifiers,” *Appl. Soft Comput.*, vol. 50, pp. 135–141, Jan. 2017.

[52] M. M. ohammadian Iman Makhani, “RFM-Based customer segmentation as an elaborative analytical tool for enriching the creation of sales and trade marketing strategies,” *Int. Acad. J. Account. Financ. Manag.*, vol. 3, pp. 21–35, Jan. 2016.

[53] R. He and J. McAuley, “Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering,” in *Proceedings of the 25th International Conference on World Wide Web*, pp. 507–517, 2016.

[54] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, “Image-Based Recommendations on Styles and Substitutes,” in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52, 2015.

[55] X. Zhang, J. Zhao, and Y. LeCun, “Character-level Convolutional Networks for Text Classification,” pp. 1–9, 2015.

[56] J.-T. Wei, S.-Y. Lin, and H.-H. Wu, “A review of the application of RFM model,” *African J. Bus. Manag. Victoria Isl.*, vol. 4, no. 19, pp. 4199–4206, Dec. 2010.

[57] F. A. Pozzi, E. Fersini, E. Messina, and B. Liu, “Chapter 1: Challenges of Sentiment Analysis in Social Networks: An Overview,” in *Sentiment Analysis in Social Networks*, Boston: Morgan Kaufmann, pp. 1–11, 2017.

[58] “Automatic natural language processing based data extraction,” Feb. 2016.

[59] A. Pinto, H. Gonçalo Oliveira, and A. Oliveira Alves, “Comparing the

Performance of Different NLP Toolkits in Formal and Social Media Text," *DROPS-IDN/6008*, 2016.

[60] E. Kontopoulos, C. Berberidis, T. Dergiades, and N. Bassiliades, "Ontology-based sentiment analysis of twitter posts," *Expert Syst. Appl.*, vol. 40, no. 10, pp. 4065–4074, Aug. 2013.

[61] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment strength detection for the social web," *J. Am. Soc. Inf. Sci. Technol.*, vol. 63, no. 1, pp. 163–173, Jan. 2012.

[62] C. Tang and L. Guo, "Digging for gold with a simple tool: Validating text mining in studying electronic word-of-mouth (eWOM) communication," *Mark. Lett.*, vol. 26, no. 1, pp. 67–80, Mar. 2015.

[63] A. Majumdar and I. Bose, "Do tweets create value? A multi-period analysis of Twitter use and content of tweets for manufacturing firms," *Int. J. Prod. Econ.*, vol. 216, no. April, pp. 1–11, 2019.

[64] H.-C. Chang and C.-Y. Wang, "E-Memory Choice Architecture," *Int. J. Online Mark.*, vol. 9, no. 1, pp. 24–37, 2018.

[65] J. S. Deshmukh and A. Kumar, "Applied Computing and Informatics Entropy based classifier for cross-domain opinion mining," *Appl. Comput. Informatics*, vol. 14, no. 1, pp. 55–64, 2018.

[66] D. Hazarika, S. Poria, P. Vij, G. Krishnamurthy, E. Cambria, and R. Zimmermann, "Modeling Inter-Aspect Dependencies for Aspect-Based Sentiment Analysis," pp. 266–270, 2018.

[67] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 5876–5883, 2018.

[68] D. Agrawal and C. Aggarwal, "On the Design and Quantification of Privacy Preserving Data Mining Algorithms," in *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems -*

PODS '01, pp. 247–255, 2001.

[69] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy, “Mining Data Streams : A Review,” vol. 34, no. 2, pp. 18–26, 2005.

[70] J. Org, “Spatial Data Mining : Database Primitives , Algorithms and Efficient DBMS Support,” pp. 193–216, 2000.

[71] M. Tsoy and V. Shchekoldin, “RFM-analysis as a tool for segmentation of high-tech products ' consumers,” in *2016 13th International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, pp. 290–293, 2016.

[72] R. A. I. T. Daoud, “Combining RFM Model and Clustering Techniques for Customer Value Analysis of a Company selling online,” in *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*, pp. 1–6, 2015.

[73] C. Naik and A. Kharwar, “Knowledge Discovery of Weighted RFM-QR Sequential Patterns With Multi Time Interval From Customer Sequence Database,” in *2015 International Conference on Computer, Communication and Control (IC4) Publisher: IEEE*, pp. 1–8, 2015.

[74] “Apache OpenNLP.” [Online]. Available: https://opennlp.apache.org/. [Accessed: 29-Mar-2018].

[75] “Apache Lucene - Welcome to Apache Lucene.” [Online]. Available: https://lucene.apache.org/. [Accessed: 29-Mar-2018].

[76] C. Y. Suen, “n-Gram Statistics for Natural Language Understanding and Text Processing,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 164–172, Apr. 1979.

[77] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” *Lingvisticae Investig.*, vol. 30, no. 1, pp. 3–26, Aug. 2007.

[78] D. Gruhl, R. Guha, R. Kumar, J. Novak, and A. Tomkins, “The predictive power of online chatter,” in *Proceeding of the eleventh ACM SIGKDD international*

conference on Knowledge discovery in data mining - KDD ’05, p. 78, 2005.

[79] B. Liu, M. Hu, and J. Cheng, “Opinion Observer: Analyzing and Comparing Opinions on the Web,” in *Proceedings of the 14th International Conference on World Wide Web*, pp. 342–351, 2005.

APPENDIX A
PUBLICATIONS

- M. Zaw and P. Tandayya, “Multi-level Sentiment Information Extraction Using the CRbSA Algorithm,” in *2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2018, pp. 1–6.
- M. Zaw and P. Tandayya, “Aspect-based and Multi-level Sentiment Information Applying Contrast Dictionary,” submitted to International Journal of Information Systems and Management (IJISAM) on 15 November 2019.
  https://www.inderscience.com/jhome.php?jcode=ijisam
- M. Zaw and P. Tandayya, “Product Categorization for Social Marketing Applying the RFC Model and Data Science Techniques,” submitted to International Journal of Business Analytics (IJBAN) on 11 November 2019.
  https://www.igi-global.com/journal/international-journal-business-analytics/67141#indices

APPENDIX B
LITS OF CONTRAST WORDS

but

however

in contrast

while

yet

differ

difference

despite

Variation

still

contrary

conversely

otherwise

other hand

nevertheless

even so

alternatively

though

instead

nonetheless

even though

in spite

in comparison

although

notwithstanding

rather

differently

unlike

meanwhile

whereas

APPENDIX C

IMPORTANT CODES

1. Contrast Dictionary or Multi-level Sentiment Analysis

```
public class Contract_Test {
    private static String TokenModelFile = "C:\\Users\\Zaw\\Dropbox\\opinion-lexicon-English\\en-token.bin";
    private static String PositiveFile = "C:\\Users\\Zaw\\Dropbox\\opinion-lexicon-English\\Positive_words1.txt";
    private static String NegativeFile = "C:\\Users\\Zaw\\Dropbox\\opinion-lexicon-
English\\SA_Words_List\\Negative_words2.txt";
    private static String NegativeWordsFile = "C:\\Users\\Zaw\\Dropbox\\opinion-lexicon-
English\\SA_Words_List\\Negative_Words.txt";
    private static String ContrastFile = "C:\\Users\\Zaw\\Dropbox\\opinion-lexicon-
English\\SA_Words_List\\Contrast_Words.txt";

    public static void main(String arg[]) throws IOException, FileNotFoundException, SQLException, ClassNotFoundException {
        String text = "This is cheap and it feels cheap  but it works and thats all that matters. I have one on the floor and two on the
ceiling for months.";
//        String text = "bought these for my k2500 suburban plenty of length front to rear.,  i ended up using them right out of the
box, i for a dead battery in my truck.,  they worked great.,  several people have complained about the clamps.,  they worked well
for me, i had no issues with poor connections.,  cables are well made.";
        System.out.println("text is : " + text);
        ArrayList<String[]> tokenset = Tokens(text);
        ArrayList<String[]> SW_list = Aft_StopWord(tokenset);
        Final_Opinion(SW_list);
    }

    public static ArrayList<String[]> Aft_StopWord(ArrayList<String[]> tokenset) throws IOException {
        ArrayList<String[]> SW_list = new ArrayList();
        for (String[] s : tokenset) {
//            System.out.println("tokenset is : " + Arrays.toString(s));
            String[] SW = Remove_StopWords(s);
            SW_list.add(SW);
        }
        return SW_list;
    }

    public static String[] Remove_StopWords(String[] tokenTest) throws IOException {
        //Read the Stop Words file
        Set<String> SW_list = phraseDictionary("C:\\Users\\Zaw\\Dropbox\\opinion-lexicon-
English\\SA_Words_List\\SW_without_neg.txt");
        String[] stopWords = SW_list.toArray(new String[SW_list.size()]);

        ArrayList<String> wordsList = new ArrayList<String>();
        for (String b : tokenTest) {
            wordsList.add(b);
        }

        for (int i = 0; i < wordsList.size(); i++) {
            // get the item as string
            for (int j = 0; j < stopWords.length; j++) {
                if (wordsList.contains(stopWords[j])) {
                    wordsList.remove(stopWords[j]);//remove it
                }
            }
        }

        List<String> list = new ArrayList<>();
        for (String str : wordsList) {
            list.add(str);
//            System.out.print(str + ", ");
        }
        String[] remove_sw_array = list.toArray(new String[list.size()]);
        return remove_sw_array;
    }

    //Calculate for three sentiments
    public static String average(List<String> opinions) {
        String sentiment = opinions.toString();
        String[] sent_array = sentiment.split(",");
        int pos_count = count(sentiment, "Positive");
        int neg_count = count(sentiment, "Negative");
        System.out.println("pos_count is : " + pos_count);
        System.out.println("neg_count is : " + neg_count);
//        System.out.println("Total is : " + sent_array.length);
        String final_opinion = null;
        float pos_avg = 100 * ((float) pos_count / (float) (pos_count + neg_count));
        System.out.println("pos_avg is : " + pos_avg);
        float neg_avg = 100 * ((float) neg_count / (float) (pos_count + neg_count));
//        System.out.println("neg_avg is : " + neg_avg);
        if (pos_avg > 49) {
            final_opinion = "1";
        } else {
            final_opinion = "-1";
        }
        return final_opinion;
    }

    public static List<String> Cont_level(ArrayList<String> sent_level) {
        String[] sent = new String[sent_level.size()];
        sent = sent_level.toArray(sent);
        System.out.println("Before check contrast : " + Arrays.toString(sent));
        String cont_check = null;
        for (int i = 1; i < sent.length; i++) {
            if (sent[i] == "Contrast") {
                if (sent[i - 1] == "Positive" && sent[i + 1] == "Positive") {
                    cont_check = "Positive";
                } else if (sent[i - 1] == "Positive" && sent[i + 1] == "Negative") {
                    cont_check = "Negative";
                } else if (sent[i - 1] == "Negative" && sent[i + 1] == "Negative") {
                    cont_check = "Negative";
                } else if (sent[i - 1] == "Negative" && sent[i + 1] == "Positive") {
                    cont_check = "Positive";
                } else {
                    cont_check = "Neutral";
                }
                sent[i] = cont_check;
                sent[i - 1] = null;
                sent[i + 1] = null;
            }
        }
        System.out.println("After check contrast  : " + Arrays.toString(sent));
        List<String> ct_st = new ArrayList<>();
        for (String s : sent) {
            if (s != null && s != "Neutral" && s.length() > 0) {
                ct_st.add(s);
            }
        }
        System.out.println("Sentiment sentences in the whole feedback : " + ct_st.toString());
        return ct_st;
    }

    public static ArrayList<String> Sentence_sent(ArrayList<String[]> SW_list) throws IOException {
        Set<String> positive = phraseDictionary(PositiveFile);
        Set<String> negative = phraseDictionary(NegativeFile);
        Set<String> neg_words = phraseDictionary(NegativeWordsFile);
        Set<String> contract = phraseDictionary(ContrastFile);
//        System.out.println("check text is: " + Arrays.toString(text));
        ArrayList<String> sentences_contrast_sentiment = new ArrayList<>();
        ArrayList<String> evaluate_data = new ArrayList<>();

        for (String[] b : SW_list) {
            int negativeCount = 0, positiveCount = 0;
            String neg = null;
            String senti = null;
            String trend = null;
            String cont = null;
            System.out.println("Text after removeSW is : " + Arrays.toString(b));
            for (int n = 1; n <= 2; n++) {
                for (String ngram : ngrams(n, b)) {
                    if (neg_words.contains(ngram)) {
                        System.out.println("Found neg_words " + negativeCount + ":" + ngram);
                        neg = "Negative";
                    } else if (positive.contains(ngram)) {
                        System.out.println("Found positive " + positiveCount + ":" + ngram);
                        positiveCount++;
                        senti = "Positive";
                    } else if (negative.contains(ngram)) {
                        System.out.println("Found negative " + negativeCount + ":" + ngram);
                        negativeCount++;
                        senti = "Negative";
                    }
                    if (contract.contains(ngram)) {
                        System.out.println("Found contract_words : " + ngram);
                        cont = "Contrast";
                        sentences_contrast_sentiment.add(cont);
                    }
                }
            }
            if (neg == "Negative") {
                if (senti == "Positive") {
                    positiveCount--;
                    negativeCount++;
//                  System.out.println("By checking negative word, it sentence is : Negative");
                } else if (senti == "Negative") {
                    positiveCount++;
                    negativeCount--;
//                  System.out.println("By checking negative word, it sentence is : Positive");
                }
            }
            System.out.println("Final positive count is : " + positiveCount);
            System.out.println("Final Negative count is : " + negativeCount);
            if (positiveCount > negativeCount) {
                trend = "Positive";
                System.out.println("Category is : positive");
                sentences_contrast_sentiment.add(trend);
            } else if (positiveCount < negativeCount) {
                trend = "Negative";
                System.out.println("Category is : Negative");
                sentences_contrast_sentiment.add(trend);
            } else {
                trend = "Neutral";
                System.out.println("Category is : Neutral");
                sentences_contrast_sentiment.add(trend);
            }
            evaluate_data.add(trend);
            evaluate_data.add(Arrays.toString(b));
        }
        System.out.println("evaluate_data is : " + evaluate_data.toString());
        System.out.println("Sentiment sentences in the whole feedback : " + sentences_contrast_sentiment.toString());
        return sentences_contrast_sentiment;
    }

    public static ArrayList<String[]> Tokens(String text) throws FileNotFoundException, IOException, SQLException,
ClassNotFoundException {
        String[] txt = text.split(",");

        // Read Tokenizer model
        InputStream inputStream1 = new FileInputStream(TokenModelFile);
        TokenizerModel tokenModel = new TokenizerModel(inputStream1);

        //Instantiating the TokenizerME class
        TokenizerME tokenizer = new TokenizerME(tokenModel);
        ArrayList<String[]> tokenset = new ArrayList<>();
        for (String s : txt) {
            String[] tokens = tokenizer.tokenize(s);
            tokenset.add(tokens);
        }
        return tokenset;
    }

    public static String Final_Opinion(ArrayList<String[]> SW_list) throws IOException {
        ArrayList<String> sentences_contrast_sentiment = Sentence_sent(SW_list);
        List<String> cont = Cont_level(sentences_contrast_sentiment);
//        System.out.println("Cont Check is : " + cont.toString());
        String opinion = average(cont);
        System.out.println("Final opinion is : " + opinion);
        return opinion;
    }

    public static Set<String> phraseDictionary(String fileName) throws IOException {
        Set<String> words = new HashSet<String>();
        BufferedReader br = new BufferedReader(new FileReader(fileName));
        String line;
        while ((line = br.readLine()) != null) {
            words.add(line);
        }
        br.close();
        return words;
    }

    public static List<String> ngrams(int n, String[] words) {
        List<String> ngrams = new ArrayList<String>();
//        String[] words = str.split(" ");
        for (int i = 0; i < words.length - n + 1; i++) {
            ngrams.add(concat(words, i, i + n));
        }
        return ngrams;
    }

    public static String concat(String[] words, int start, int end) {
        StringBuilder sb = new StringBuilder();
        for (int i = start; i < end; i++) {
            sb.append((i > start ? " " : "") + words[i]);
        }
        return sb.toString();
    }
}
```

2. Aspect-level Sentiment Analysis

```
public class asp_levl_sents_extert_test_ {
    private static String TokenModelFile = "C:\\Users\\Zaw\\Dropbox\\opinion-lexicon-English\\en-token.bin";
    private static String PositiveFile = "C:\\Users\\Zaw\\Dropbox\\opinion-lexicon-English\\Positive_words1.txt";
    private static String NegativeFile = "C:\\Users\\Zaw\\Dropbox\\opinion-lexicon-
English\\SA_Words_List\\Negative_words2.txt";
    private static String NegativeWordsFile = "C:\\Users\\Zaw\\Dropbox\\opinion-lexicon-
English\\SA_Words_List\\Negative_Words.txt";

    private static String asp_prod = "C:\\Users\\Zaw\\Desktop\\Aspect_Test_Temp_file\\product_aspects.txt";
    private static String asp_qual = "C:\\Users\\Zaw\\Desktop\\Aspect_Test_Temp_file\\quality_aspects.txt";
    private static String durab_qual = "C:\\Users\\Zaw\\Desktop\\Aspect_Test_Temp_file\\durability_aspects.txt";
    private static String asp_recom = "C:\\Users\\Zaw\\Desktop\\Aspect_Test_Temp_file\\recommend_aspects.txt";
    private static String asp_cost = "C:\\Users\\Zaw\\Desktop\\Aspect_Test_Temp_file\\cost_aspects.txt";
    private static String asp_usag = "C:\\Users\\Zaw\\Desktop\\Aspect_Test_Temp_file\\usage_aspects.txt";
    private static String asp_app_time = "C:\\Users\\Zaw\\Desktop\\Aspect_Test_Temp_file\\app_time_aspects.txt";

    public static void main(String arg[]) throws SQLException, ClassNotFoundException, IOException {
        String select_sentences_table = "aspect_extract";
        String insert_sentences_table = "aspect_sentiments_ext";

        ResultSet select = selectdb(select_sentences_table);
        while (select.next()) {
            String x = select.getString("Feedback");
            String[] p_asp_ext = product_condition_named_enterities(x);
            String[] qual_asp_ext = quality_named_enterities(x);
            String[] durab_asp_ext = durability_named_enterities(x);
            String[] recom_asp_ext = recommend_named_enterities(x);
            String[] cost_asp_ext = cost_named_enterities(x);
            String[] usag_asp_ext = usage_named_enterities(x);
            String[] app_time_asp_ext = working_time_named_enterities(x);
            ArrayList<String[]> tokenset = Tokens(x);
            ArrayList<String[]> SW_list = Aft_StopWord(tokenset);

            Set<String> prods = phraseDictionary(asp_prod);
            Set<String> qual = phraseDictionary(asp_qual);
            Set<String> durab = phraseDictionary(durab_qual);
            Set<String> recom = phraseDictionary(asp_recom);
            Set<String> cost = phraseDictionary(asp_cost);
            Set<String> usg = phraseDictionary(asp_usag);
            Set<String> app_time = phraseDictionary(asp_app_time);
            Set<String> positive = phraseDictionary(PositiveFile);
            Set<String> negative = phraseDictionary(NegativeFile);
            Set<String> neg_words = phraseDictionary(NegativeWordsFile);

            ArrayList<String> sentences_aspect_sentiment = new ArrayList<>();
            ArrayList<String> product_aspect_sentiment = new ArrayList<>();
            ArrayList<String> quality_aspect_sentiment = new ArrayList<>();
            ArrayList<String> durability_aspect_sentiment = new ArrayList<>();
            ArrayList<String> recommend_aspect_sentiment = new ArrayList<>();
            ArrayList<String> cost_aspect_sentiment = new ArrayList<>();
            ArrayList<String> usage_aspect_sentiment = new ArrayList<>();
            ArrayList<String> app_time_aspect_sentiment = new ArrayList<>();
            String asp = null;
            int i = 1;

            for (String[] b : SW_list) {
                System.out.println("Token sets sentencent : " + Arrays.toString(b));
                int negativeCount = 0, positiveCount = 0;
                String neg = null;
                String senti = null;
                String trend = null;
                String prod_asps = null;
                String qual_asps = null;
                String durab_asps = null;
                String recom_asps = null;
                String cst_asps = null;
                String usg_asps = null;
                String ap_t_asps = null;
                for (int n = 1; n <= 3; n++) {
```

```
    for (String ngram : ngrams(n, b)) {
        if (prods.contains(ngram)) {
            prod_asps = "Product";
        } else if (qual.contains(ngram)) {
            qual_asps = "Quality";
        } else if (durab.contains(ngram)) {
            durab_asps = "Durability";
        } else if (recom.contains(ngram)) {
            recom_asps = "Recommend";
        } else if (cost.contains(ngram)) {
            cst_asps = "Cost";
        } else if (usg.contains(ngram)) {
            ap_t_asps = "Usage";
        } else if (app_time.contains(ngram)) {
            usg_asps = "Time";
        } else if (neg_words.contains(ngram)) {
            System.out.println("Found neg_words " + negativeCount + ":" + ngram);
            neg = "Negative";
        } else if (positive.contains(ngram)) {
            System.out.println("Found positive " + positiveCount + ":" + ngram);
            positiveCount++;
            senti = "Positive";
        } else if (negative.contains(ngram)) {
            System.out.println("Found negative " + negativeCount + ":" + ngram);
            negativeCount++;
            senti = "Negative";
        }
    }
}
if (neg == "Negative") {
    if (senti == "Positive") {
        positiveCount--;
        negativeCount++;
    } else if (senti == "Negative") {
        positiveCount++;
        negativeCount--;
    }
}
System.out.println("Final positive count is : " + positiveCount);
System.out.println("Final Negative count is : " + negativeCount);
if (positiveCount > negativeCount) {
    trend = "Positive";
    System.out.println("Category is : positive");
    sentences_aspect_sentiment.add(trend);
} else if (positiveCount < negativeCount) {
    trend = "Negative";
    System.out.println("Category is : Negative");
    sentences_aspect_sentiment.add(trend);
} else {
    trend = "Neutral";
    System.out.println("Category is : Neutral");
    sentences_aspect_sentiment.add(trend);
}

if (prod_asps == "Product") {
    asp = trend;
    System.out.println("asp : " + asp);
    product_aspect_sentiment.add(asp);
} else if (qual_asps == "Quality") {
    asp = trend;
    quality_aspect_sentiment.add(asp);
} else if (durab_asps == "Durability") {
    asp = trend;
    durability_aspect_sentiment.add(asp);
} else if (recom_asps == "Recommend") {
    asp = trend;
    recommend_aspect_sentiment.add(asp);
} else if (cst_asps == "Cost") {
    asp = trend;
    cost_aspect_sentiment.add(asp);
} else if (usg_asps == "Usage") {
```

```
          asp = trend;
          usage_aspect_sentiment.add(asp);
        } else if (ap_t_asps == "Time") {
          asp = trend;
          app_time_aspect_sentiment.add(asp);
        }
      }
      InsertDB(insert_sentences_table, select.getString("contrast"), select.getString("word_count"),
select.getString("senti_strength"), select.getString("product"), product_aspect_sentiment.toString(), select.getString("durability"),
durability_aspect_sentiment.toString(), select.getString("quality"), quality_aspect_sentiment.toString(),
select.getString("recommend"), recommend_aspect_sentiment.toString(), select.getString("cost"),
cost_aspect_sentiment.toString(), select.getString("usg"), usage_aspect_sentiment.toString(), select.getString("working_time"),
app_time_aspect_sentiment.toString(), select.getString("Feedback"));
    }
  }
}
```

3. RFC Testing on Automotive Products

```
public class RFC_testing {
    public static void main(String[] args) throws ClassNotFoundException, SQLException, IOException {
        String table = "Automotive";
        Credit_Score(table);
        Frequency_Score(table);
        Recency_Score(table);
    }

    public static ResultSet Credit_Select(String table_name) throws ClassNotFoundException, SQLException {
        Connection conn = connectdb();
        Statement st = (Statement) conn.createStatement();
        String sql = "SELECT productID, SUM(`credit`) AS c FROM " + table_name + " GROUP BY productID HAVING c > 0
ORDER BY c ASC";
        st.executeQuery(sql);
        ResultSet rs = st.executeQuery(sql);
        System.out.println(" SUCCESS! select db\n");
        return rs;
    }

    public static String[] Credit_Score(String table_name) throws ClassNotFoundException, SQLException, IOException {
        ResultSet rs = Credit_Select(table_name);
        StringBuilder sb = new StringBuilder();
        List<String> pID = new ArrayList<String>();

        while (rs.next()) {
            pID.add(rs.getString("productID"));
            if (rs.getString("c") != null) {
                sb.append(rs.getString("c")).append(" ");
            }
        }

        List<String> C_value = Arrays.asList(sb.toString().split(" "));
        String[] C_count = (String[]) C_value.toArray();

        String[] Score = new String[C_value.toArray().length];
        String[] Credit_State = new String[C_value.toArray().length];
        int j = 0;
        for (int i = 1; i <= C_value.toArray().length; i++) {
            float Level = Quintile_Equ(count(sb.toString(), C_count[j]), i, C_value.toArray().length);
            if (Level < 1.5) {
                Score[j] = "1";
                Credit_State[j] = "Low";
            } else if (Level >= 1.5 && Level < 2.5) {
                Score[j] = "2";
                Credit_State[j] = "Low";
            } else if (Level >= 2.5 && Level < 3.5) {
                Score[j] = "3";
                Credit_State[j] = "Low";
            } else if (Level >= 3.5 && Level < 4.5) {
                Score[j] = "4";
                Credit_State[j] = "High";
            } else {
                Score[j] = "5";
                Credit_State[j] = "High";
            }
            j++;
        }

        List<String> Score_list = new ArrayList<>();
        for (String s : Score) {
            if (s != null && s.length() > 0) {
                Score_list.add(s);
            }
        }

        List<String> Credit_list = new ArrayList<>();
        for (String p : Credit_State) {
            if (p != null && p.length() > 0) {
                Credit_list.add(p);
            }
        }

        //Store the values set in arrayList
        String[] pID_F_Score_Array = new String[Score.length];
        ArrayList<String> pID_F_Score = new ArrayList<>();
        for (int b = 0; b < Score.length; b++) {
            pID_F_Score.add(pID.get(b));
            pID_F_Score.add(Credit_list.get(b));
            Update_Credit_toDB(table_name, pID.get(b), Credit_list.get(b));
            pID_F_Score_Array = pID_F_Score.toArray(new String[pID_F_Score.size()]);
        }
        return pID_F_Score_Array;
    }

    public static void Update_Credit_toDB(String table_name, String a, String b) throws SQLException,
ClassNotFoundException, IOException {
        Connection conn = connectdb();
        Statement st = (Statement) conn.createStatement();
        String updateTableSQL = "UPDATE " + table_name + " SET `C`='" + b + "' where `productID` = '" + a + "'";
        st.executeUpdate(updateTableSQL);
        System.out.println(" SUCCESS! update data db\n");
        conn.close();
    }

    public static ResultSet Frequency_Select(String table_name) throws ClassNotFoundException, SQLException {
        Connection conn = connectdb();
        Statement st = (Statement) conn.createStatement();
        String sql = "SELECT productID, COUNT(*) AS c FROM " + table_name + " GROUP BY productID HAVING c > 0
ORDER BY c ASC";
        st.executeQuery(sql);
        ResultSet rs = st.executeQuery(sql);
        System.out.println(" SUCCESS! select db\n");
        return rs;
    }

    public static ResultSet Recency_Select(String table_name) throws ClassNotFoundException, SQLException {
        Connection conn = connectdb();
        Statement st = (Statement) conn.createStatement();
        String sql = "SELECT productID, MIN(diffDate) FROM " + table_name + " GROUP BY productID ORDER BY
MIN(diffDate) DESC";
        st.executeQuery(sql);
        ResultSet rs = st.executeQuery(sql);
        System.out.println(" SUCCESS! select db\n");
        return rs;
    }

    public static String[] Frequency_Score(String table_name) throws ClassNotFoundException, SQLException, IOException {
        ResultSet rs = Frequency_Select(table_name);
        StringBuilder sb = new StringBuilder();
        List<String> pID = new ArrayList<String>();

        while (rs.next()) {
            pID.add(rs.getString("productID"));
            if (rs.getString("c") != null) {
                sb.append(rs.getString("c")).append(" ");
            }
        }

        List<String> F_value = Arrays.asList(sb.toString().split(" "));
        String[] F_count = (String[]) F_value.toArray();

        String[] Score = new String[F_value.toArray().length];
        String[] Population_State = new String[F_value.toArray().length];
        int j = 0;
        for (int i = 1; i <= F_value.toArray().length; i++) {
            float Level = Quintile_Equ(count(sb.toString(), F_count[j]), i, F_value.toArray().length);
            if (Level < 1.5) {
                Score[j] = "1";
                Population_State[j] = "Low";
            } else if (Level >= 1.5 && Level < 2.5) {
                Score[j] = "2";
                Population_State[j] = "Low";
            } else if (Level >= 2.5 && Level < 3.5) {
                Score[j] = "3";
                Population_State[j] = "Low";
            } else if (Level >= 3.5 && Level < 4.5) {
                Score[j] = "4";
                Population_State[j] = "High";
            } else {
                Score[j] = "5";
                Population_State[j] = "High";
            }
            j++;
        }

        List<String> Score_list = new ArrayList<>();
        for (String s : Score) {
            if (s != null && s.length() > 0) {
                Score_list.add(s);
            }
        }

        List<String> Population_list = new ArrayList<>();
        for (String p : Population_State) {
            if (p != null && p.length() > 0) {
                Population_list.add(p);
            }
        }

        String[] pID_F_Score_Array = new String[Score.length];
        ArrayList<String> pID_F_Score = new ArrayList<>();
        for (int b = 0; b < Score.length; b++) {
            pID_F_Score.add(pID.get(b));
            pID_F_Score.add(Population_list.get(b));
            Update_frequency_toDB(table_name, pID.get(b), Population_list.get(b));
            pID_F_Score_Array = pID_F_Score.toArray(new String[pID_F_Score.size()]);
        }
        return pID_F_Score_Array;
    }

    public static void Update_frequency_toDB(String table_name, String a, String b) throws SQLException,
ClassNotFoundException, IOException {
        Connection conn = connectdb();
        Statement st = (Statement) conn.createStatement();
        String updateTableSQL = "UPDATE " + table_name + " SET `F`='" + b + "' where `productID` = '" + a + "'";
        st.executeUpdate(updateTableSQL);
        System.out.println(" SUCCESS! update data db\n");
        conn.close();
    }

    public static String[] Recency_Score(String table_name) throws ClassNotFoundException, SQLException, IOException {
        ResultSet rs = Recency_Select(table_name);
        StringBuilder sb = new StringBuilder();
        List<String> pID = new ArrayList<>();
        while (rs.next()) {
            pID.add(rs.getString("productID"));
            System.out.println(rs.getString("productID") + "\t" + "Recency difference days are : " + rs.getString("MIN(diffDate)"));
            if (!"".equals(rs.getString("MIN(diffDate)"))) {
                sb.append(rs.getString("MIN(diffDate)")).append(" ");
            }
        }
        System.out.println("Sb is : " + sb);
        List<String> list = Arrays.asList(sb.toString().split(" "));
        String[] R_count = (String[]) list.toArray();

        String[] Score = new String[list.toArray().length];
        String[] Active_State = new String[list.toArray().length];
        int j = 0;
        for (int i = 1; i <= list.toArray().length; i++) {
            float Level = Quintile_Equ(count(sb.toString(), R_count[j]), i, list.toArray().length);
            if (Level < 1.5) {
                Score[j] = "1";
                Active_State[j] = "Low";
            } else if (Level >= 1.5 && Level < 2.5) {
                Score[j] = "2";
                Active_State[j] = "Low";
            } else if (Level >= 2.5 && Level < 3.5) {
                Score[j] = "3";
                Active_State[j] = "Low";
            } else if (Level >= 3.5 && Level < 4.5) {
                Score[j] = "4";
                Active_State[j] = "High";
            } else {
                Score[j] = "5";
                Active_State[j] = "High";
            }
            j++;
        }

        List<String> Score_list = new ArrayList<>();
        for (String s : Score) {
            if (s != null && s.length() > 0) {
                Score_list.add(s);
            }
        }
        Score = Score_list.toArray(new String[Score_list.size()]);

        List<String> Active_list = new ArrayList<>();
        for (String p : Active_State) {
            if (p != null && p.length() > 0) {
                Active_list.add(p);
            }
        }

        //Store the values set in arrayList
        String[] pID_Active_Array = new String[Score.length];
        ArrayList<String> pID_A_Score = new ArrayList<>();
        for (int b = 0; b < Score.length; b++) {
            pID_A_Score.add(pID.get(b));
            pID_A_Score.add(Active_list.get(b));
            Update_recency_toDB(table_name, pID.get(b), Active_list.get(b));
        }
        System.out.println("pID_A_Score is : " + pID_A_Score.toString());
        pID_Active_Array = pID_A_Score.toArray(new String[pID_A_Score.size()]);
        System.out.println("pID_F_Score_Array is : " + Arrays.toString(pID_Active_Array));
        return pID_Active_Array;
    }

    public static void Update_recency_toDB(String table_name, String a, String b) throws SQLException,
ClassNotFoundException, IOException {
//          String a = "B00002243X", b = "ssssss";
        Connection conn = connectdb();
        Statement st = (Statement) conn.createStatement();
        // UPDATE `relation_variables` SET `Popularity`="a"

        String updateTableSQL = "UPDATE " + table_name + " SET `R`='" + b + "' where `productID` = '" + a + "'";
        st.executeUpdate(updateTableSQL);
        System.out.println(" SUCCESS! update data db\n");
        conn.close();
    }

    public static float Quintile_Equ(int Equal, int Posit, int Total) {
        float a = (float) 0.5;
        float c = a * Equal;
//          System.out.println(c);
        float d = Posit + c;
//          System.out.println(d);
        int e = 5;
        float f = e * d;
//          System.out.println(f);
        float result = f / Total;
//          System.out.println(result);
```

```
        return result;
    }

    public static int count(String string, String substr) {
        int i;
        int last = 0;
        int count = 0;
        do {
            i = string.indexOf(substr, last);
            if (i != -1) {
                count++;
            }
            last = i + substr.length();
        } while (i != -1);
        return count;
    }
}
```

APPENDIX D
VITAE

Name Mr. Myint Zaw

Student ID 5910130012

Education Attainment

Degree	Name of Institution	Year of Graduation
Bachelor of Computer Technology	University of Computer Study Yangon (UCSY)	2009
Bachelor of Computer Technology (Honours)	University of Computer Study Yangon (UCSY)	2011
Master of Computer Technology	University of Computer Study Yangon (UCSY)	2013

Scholarship Awards during Enrolment

Higher Education Research Promotion and the Thailand's Education Hub for Southern Region of ASEAN Countries Project Office of the Higher Education Commission.

List of Publication and Proceeding

- M. Zaw and P. Tandayya, "Multi-level Sentiment Information Extraction Using the CRbSA Algorithm," in *2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2018, pp. 1–6.
- M. Zaw and P. Tandayya, "Aspect-based and Multi-level Sentiment Information Applying Contrast Dictionary," submitted to International Journal of Information Systems and Management (IJISAM) on 25 November 2019. https://www.inderscience.com/jhome.php?jcode=ijisam
- M. Zaw and P. Tandayya, "Product Categorization for Social Marketing Applying the RFC Model and Data Science Techniques," submitted to International Journal of Business Analytics (IJBAN) on 11 November 2019. https://www.igi-global.com/journal/international-journal-business-analytics/67141