

**Sentiment Analysis on Movie Rating Using N-Gram Model**

By

**Syed Akmal Bin Syed Othman (11003)**

**Dissertation submitted in partial fulfillment of**

**the requirement for the**

**Bachelor of Information and Communication Technology (Hons)**

**JANUARY 2011**

**Universiti Teknologi Petronas**

**Bandar Seri Iskandar**

**31750 Tronoh**

**Perak Darul Redzuan**

# CERTIFICATION OF APPROVAL

## **Sentiment Analysis on Movie Rating Using N-Gram Model**

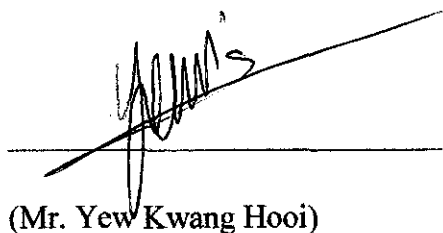
by

Syed Akmal bin Syed Othman

11003

A project dissertation submitted to the  
Information Technology Programme  
University Teknologi PETRONAS  
in partial fulfillment of the requirement for the  
BACHELOR OF TECHNOLOGY (Hons)  
(INFORMATION TECHNOLOGY)

Approved by,



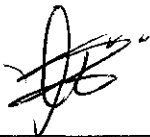
(Mr. Yew Kwang Hooi)

UNIVERSITI TEKNOLOGI PETRONAS  
TRONOH, PERAK

January 2011

## CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the reference and acknowledgements, and that the original work contained herein has not been undertaken or done by unspecified sources or persons.



---

SYED AKMAL BIN SYED OTHMAN

**TABLE OF CONTENTS**

<b>No.</b>	<b>Items</b>	<b>Pages</b>
1	LIST OF FIGURE.....	-
2	ABSTRACT.....	1
3	ACKNOWLEDGEMENT	2
3	CHAPTER 1: INTRODUCTION.....	3
	• Background of Study.....	3
	• Problem Statement.....	4
	• Objectives and Scope of Study.....	4
4	CHAPTER 2: LITERATURE REVIEW.....	5
5	CHAPTER 3: METHDOLOGY.....	11
	• Software Prototyping.....	11
	• Experimental Method.....	12
	• Project Activities.....	13
6	CHAPTER 4: PROTOTYPE ANALYSIS AND DESIGN	16
7	CHAPTER 5: RESULT AND DISCUSSION.....	19
8	CHAPTER 6: CONCLUSION AND RECOMMENDATION.	25
	• Conclusion.....	25
	• Recommendation.....	26
9	REFERENCES.....	28

## **LIST OF FIGURE**

<b>No.</b>	<b>Figure</b>	<b>Page</b>
1	Software prototyping process diagram	11
2	Scientific method processes	12
3	FYP 1 Gantt Chart	13
4	FYP 2 Gantt Chart	15
5.	Input process flow chart	16
6.	Analyze process flow chart	17
7.	Range of Result	18
8.	Application's tab view	19
9.	Data Population's tab view	20
10.	Sample Application of Movie Rating	21
11.	Test Result and Accuracy Rate	21

## **ABSTRACT**

Sentiment analysis is an emerging area of text mining where it used to analyze how human feelings or opinion towards certain products or subjects such as movies in this context. Nowadays, as the social network become increasing popular, people actively expressing their opinion about something in the internet. By using sentiment analysis, company can monitor how consumer responds to their products and at the same time improve their Customer Relationship Management (CRM). This paper will explain about what sentiment analysis is and how to implement the N-Gram model in analyzing human sentiment.

## **ACKNOWLEDGEMENT**

First of all, I would like to address my highest appreciation to God for giving me enough energy and good health before, during and after completing this project.

Alhamdulillah. Not to mention, my supervisor, Mr. Yew Kwang Hooi for his brilliant insight, support and patient throughout this project even though he is currently pursuing his studies in PhD. Without his great advice and encouragement, I may have difficulties in understanding some of the concepts and related theories. Lastly, I would like to thank my family and friends because of their moral support in whatever situation. Thank you very much.

## CHAPTER 1: INTRODUCTION

### Background of Study

Today internet has increasingly become a major medium to get information about certain product. Before buying anything or in this context buying ticket to watch a movie, people will look for reviews on the internet whether it is in online forum, opinion from friends or family members (e.g in social networks such as Twitter, Facebook and blogs). However, in business point of view, customer feedbacks about their products are very important. The question is how they would know how the customer feels about their products through the internet. Some web application allows the user to give a rating from 0 to 10 about the products. Not only that, they also can leave comments. Ratings can be calculated because it is in numbers. What about comments? How can comments be calculated and present it in statistical manners? This is why there is a need in sentiment analysis.

Before that, what is sentiment? Sentiment can be defined as general thought, feeling or sense. [1] In other words, sentiment is human opinion. Opinion is really subjective. Even though there are algorithms that can calculate the sentiment, it's really hard to reads people emotion. Emotions like love, hate, sad, fear, surprise and happy are usually expresses in everyday conversation. Human like us may able to determine how other person feels based on what they are saying. But then, many a time people does not always express their feelings directly. For this project, the author only need to determine the value of the human sentiment whether it is positive, negative or neutral. Any other human emotion will not be included into the project scope.

In English language, they are many words that can be considered as positive, negative or neutral. For example, "good" is a positive word, "not" is a negative word so on and so forth. If "good" appear in a sentence, that sentence might be positive sentence. What happen if there is "not good" appear? What could be the result? This is where N-Gram model can be applied. In N-Gram analysis, the meaning of a word may change depends on what is the word occurs before the said word.



An algorithm like N-Gram can be implemented to analyze the human perception about the products (movies). Then, the result can be represented in a statistical manner.

### **Problem Statement**

There are quite a number of applications that are used to analyze human sentiment, but they do not give an accurate result (i.e. Tweetfeel, Twendz and Twitrratr).

### **Objectives & Scope of Study**

- To create a software prototype that uses N-Gram algorithm and is able to determine the sentiment on a given sentence or article (negative, positive, neutral).
- To evaluate the performance of N-Gram compared to other algorithms.
- To invent a new way or approaches in evaluating human sentiment.

## CHAPTER 2: LITERATURE REVIEW AND THEORY

This chapter will divide into two parts. In the first part, the author will walk through the current research in sentiment analysis. Then, in second part, the author will analyze on how current N-Gram being used in the industry not only in natural language processing but also in other area.

Sentiment analysis is quite young in the computer linguistic area. An early attempt to understand the human sentiment has been done by Peter Turney and Bo Pang in 2002 [4]. They try to detect the polarity of reviews whether it is positive, negative or neutral. There are two categories of textual information which are facts and opinions. Facts are objective whereas opinions or sentiments are usually subjective [4]. In analyzing sentiments, there are two level of analysis which are document-level and sentence-level analysis. This project will focus on sentence-level. Two task need to be performed in order to accomplish this analysis.

Based on a paper wrote by Bing Liu, first, a process called subjectivity classification is perform to determine the sentence whether it is objective or subjective. Then, if the sentence is subjective, one need to determine whether it is positive or negative opinion by undergoes sentence-level sentiment classification [3]. These processes is very useful because before analyzing the sentence, it can eliminate the sentence that does not have any opinion and focused on determining the opinion on what objects or subjects whether it is positive or negative. Not all reviews or comments from the user are an opinion. User may only give general statement of the movie. For example:

*"No one has reviewed this movie, well i know why because no one has probably even seen this movie.", by Scottynaar, Moviereviews.com*

This general statement does not provide any positive or negative statement. He/She only tell others that maybe not many people watching the movie. So, this kind of sentences can be eliminated/ignore.

Opinions or sentiments can be classified in to two different types. One is base type and another is comparative type [3]. Base type means the text/sentence only refer to an object itself, but however comparative type means that it try to compare and contrast between two different objects for example by using words like better, amazing than and greater.

The author already mentions some of the method that has been used to analyze human sentiment towards a specific subject. But, why we really need to analyze those opinions?

A survey conducted in October 2004 by Paul Hitlin, Research Associate, and Lee Rainie, Director [4]. Based on the survey, 26% of adult internet users have rated a product, service or person using the internet. This figure may increase significantly today because the internet users increase from time to time. Before this, a person will ask people around them before attempting to buy something. But, today the information can easily be found by searching it on the internet (e.g Google, Yahoo and Bing).

There are many applications that apply sentiment analysis concept. Because of many people use social network such as Twitter to express their feelings, a quite number of application has been built using Twitter API. Below are the examples of applications:

1. Tweetfeel [5]



Developed by Conversion. This web application will monitors the tweets (a short message) post by the Tweeter users. Then, it will determine whether the message has positive or negative feelings about any subject like movies, musician or TV shows. User only needs to enter any keyword that he/she want to look for. Then, tweetfeel will display who talk about that keyword and his opinion towards that topic (negative or positive).

## 2. Twitrratr [6]

# twitrratr

Co-founded by Beau Frusetta, Chase Granberry and Mike Luby, this application was developed to find tweets about President of United States, Barack Obama whether positive or negative, originally. But then, they make it universal which means it can be use for any subjects. Twitrratr works quite simple. They only compare the tweet message with positive and negative words. Different from tweetfeel, twitrratr have additional category which is neutral.

## 3. Twendz [7]



Developed by Waggener Edstrom Worldwide, another sentiment analysis application that used Twitter API. This application is the most advance application compare to previous Twitter application. The major difference is twendz works on real-time. Its monitor the tweet message and display the result in real-time. The user also can control the crawling speed of the application (pause, slow, normal, somewhat fast and fast). There also can be use for commercial purpose by using twendzPro. Here, user can get the result of certain product in graphical manner. Provide the user with key performance indicators that helps in business decision making.

That's some of the current research and application available on sentiment analysis field. This project will make use of N-Gram algorithm in order to analyze the human sentiment. As discuss earlier, some of the application use simple method to analyze the sentiment by comparing the sentence with positive and negative words. This is not an effective way to analyze the sentiment.

For example:

*“The movie is actually not bad”*

In this sentence, there are two negative words ‘not’ and ‘bad’. By using the simple method, the result would be negative. But in actual meaning, the review is positive about the movie. ‘not bad’ means it is good.

So, an effective way to analyze the sentence can be done by using N-Gram algorithm. Many researchers refer N-Gram as a language model not an algorithm. But, the author makes use of this algorithm as a model in order to be implemented in software prototype later on.

N-gram used previous n-1 words in sequence to predict the next word [8]. Example:

*The big red dog*

1.  $P(\text{dog}) \leftarrow \text{uni-gram (1-gram)}$
2.  $P(\text{dog}|\text{red}) \leftarrow \text{bi-gram (2-gram)}$
3.  $P(\text{dog}|\text{big red}) \leftarrow \text{tri-gram (3-gram)}$
4.  $P(\text{dog}|\text{the big red}) \leftarrow \text{quad-gram (4-gram. This gram onwards can be called n-gram)}$

N-gram algorithm uses conditional probabilities to calculate the probabilities of the word (gram). In unigram like an example 1 above shows that the probability of “dog” in the sentence “The big red dog. However, in bi-gram, conditional probability is introduced. Example 2 shows that the probability of “red” given that “dog” is in the sentence. This process continues until the last word in the sentence. So, the total probability of word sequence can be derived like this:

$$P(\text{The}) * P(\text{big}|\text{the}) * P(\text{red}|\text{the big}) * P(\text{dog}|\text{the big red})$$

By using chain rule, the typical formula would be:

$$P(w_1^n) = P(w_1)P(w_2 | w_1)P(w_3 | w_1^2) \dots P(w_n | w_1^{n-1})$$

$$= P(w_1) \prod_{k=2}^n P(w_k | w_1^{k-1})$$

$w$  = the word

$n$  = the position of the word in the sentence

N-gram model also has been used to categorize article into specific topic. This research has been done by Peter Nather in his paper N-Gram Based Text Categorization [9]. To measure its effectiveness, he makes use of recall and precession formula.

$$recall = \frac{categories\ found\ and\ correct}{total\ categories\ correct}$$

$$precision = \frac{categories\ found\ and\ correct}{total\ categories\ found}$$

Using this formula, he can calculate the performance of the algorithm that he used to categorize the article into their categories. Before he can do text categorization, He needs to build the database to store the categories information. This can be done in text clusterization phase (The articles will be group together according to some similar characteristics).

However, text categorization differs from sentiment classification [4]. This means that, the purpose of text categorization is to categorize the text by topic. The topics can variably come from the number of document or a set of category that has been defined. In contrast with sentiment, only two or three categories involved (positive, negative and neutral). Furthermore, text categorization or document categorization deals with longer text and it much more complicated. But in sentiment, only one sentence need to be analyze.

Apart from algorithm, the language itself needs to take into consideration. [10] English language already has over 900,000 words according to Global Language Monitor. This means that the author needs to create the database for those words. The algorithm that needs to be implemented later on will be able to do calculation effectively for all English words.

By using the probability formula of n-gram and combination of statistical calculation, it can be used to analysis the sentiment of movie review or rating.



## CHAPTER 3: METHODOLOGY

There are two methods that the author will use throughout this research project which are:

1. Software prototyping: Evolutionary prototyping
2. Experimental methods

### Software Prototyping: Evolutionary prototyping

One of the objectives of this project is building a software prototype. In order to accomplish that, the author use software prototyping methodology, specifically evolutionary prototyping [11]. This type of prototyping is different from throwaway where in throwaway the prototype that has been developed will be discarded if it does not meet the requirement. In evolutionary prototyping, the developer will create the core of the prototype and constantly enhance the prototype to make it better.

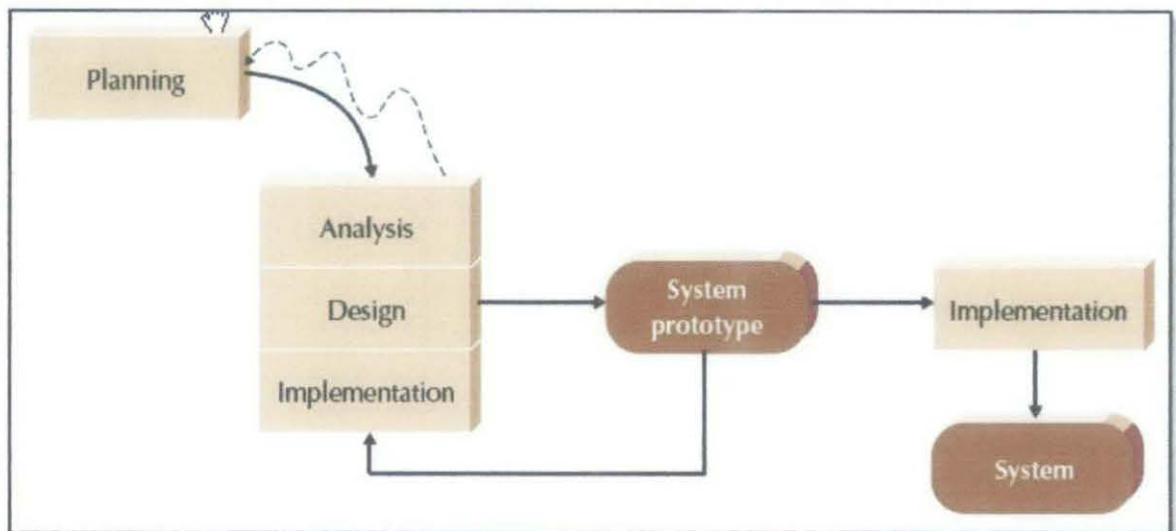


Figure 1 : Software prototyping process diagram

In this project however, the author will implement n-gram model and any other related algorithm in the prototype. After that, the prototype will be tested to see the effectiveness of the algorithm. The result of the testing is analyzed. If the result does not meet the desired result, then the prototype will be refined back.



**Scientific Methods**

Other than software prototyping, this project also uses scientific method where the author will create the hypothesis and then testing the hypothesis using the software prototype that has been developed [12]. The process flow of scientific method can summarize in following diagram.

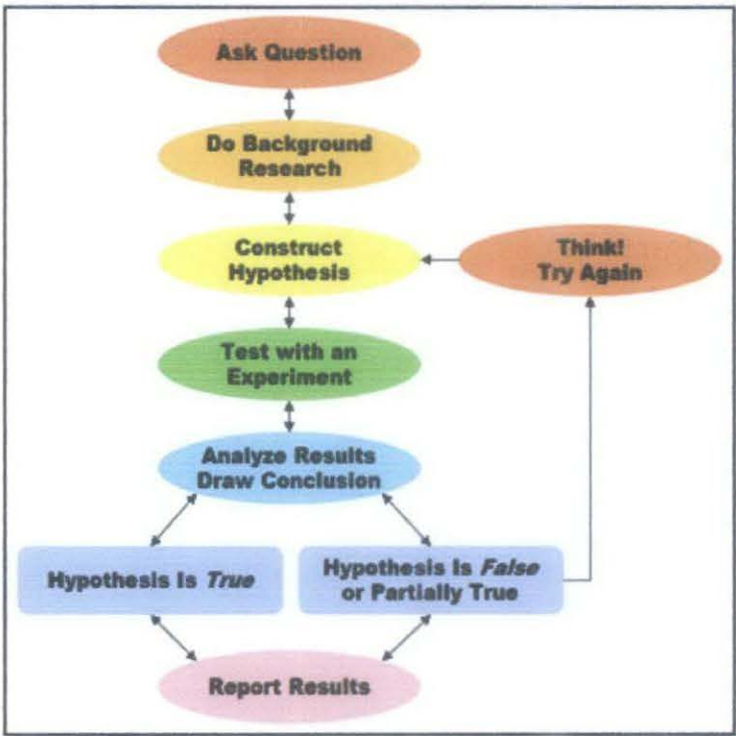


Figure 2 : Scientific method processes

Before starting an experiment of sentiment analysis using N-Gram algorithm, author make some assumption. The assumptions as follows:

- The sentence is in proper grammar. No grammar mistake involves.
- The sentence is only for single opinion from single opinion holder (user). There will be no compound opinion will included in the experiment.

### FYP 1 Project Milestones:

1. Propose project title
2. Research on project title
3. Defining scope of work
4. Research on sentiment analysis
5. Research N-Gram algorithm
6. Building prototype that can demonstrate how N-Gram works
7. Evaluating N-Gram algorithm with other method of analyzing sentiment
8. Revised the software prototype by modifying the implementation of N-Gram
9. Testing the software prototype regularly
10. Do final project documentation

NO	Detail/Week	1	2	3	4	5	6	7	8	9		10	11	12	13	14
1	Propose Project Title										MID SEM BREAK					
2	Doing General Research on Project Title															
3	Defining Scope of Work															
4	Doing Specific Research On Sentiment Analysis															
5	Doing Specific Research On N-Gram Algorithm															
6	Building Prototype That Can Demonstrate How N-Gram Works															

Figure 3 : FYP 1 Gantt chart

## **FYP 2 Project Milestones:**

1. Determine the process involves in implementing N-Gram model.
2. Design a preliminary GUI for N-Gram software prototype.
3. Implementing Unigram
  - a. Creating Unigram java class.
  - b. Creating string manipulation process for Unigram.
  - c. Create a database table that store Unigram word.
4. Implementing Bigram
  - a. Creating Bigram java class.
  - b. Create string manipulation process for Bigram.
  - c. Create a database table that store Bigram word.
5. Implementing Trigram
  - a. Creating Trigram java class.
  - b. Create string manipulation process for Trigram
  - c. Create a database table that store Trigram word.
6. Constantly updating Unigram, Bigram and Trigram table with training corpus.
7. Building hash map to store the sentiment value for Unigram, Bigram and Trigram.
8. Implement an algorithm that can score the sentiment of user input sentence.

FYP 2 Gant Chart

NO	Detail/Week	1	2	3	4	5	6	7	8		8	9	10	11	12	13	14	15
1	Determine the process involve to implement N-Gram									MID SEM BREAK								
2	Design preliminary GUI																	
3	Implementing Unigram																	
4	Implementing Bigram																	
5	Implementing Trigram																	
6	Updating Unigram, Bigram, Trigram table																	
7	Building hash map for sentiment value																	
8	Invent algorithm to score sentiment																	
9	Finalizing project																	

Figure 4 : FYP 2 Gantt chart

Tools:

- 1. Java programming language

The author most familiarize with Java programming language. So, most probably he will use this programming language do demonstrate how N-Gram works. This mainly uses to test idea and hypothesis about N-Gram and sentiment. To be able to program in Java, the author use Netbeans IDE.

# CHAPTER 4: PROTOTYPE ANALYSIS AND DESIGN

There are 2 processes involve in developing this software prototype:

- 1. Input process
- 2. Analyzing process

## 1. Input Process

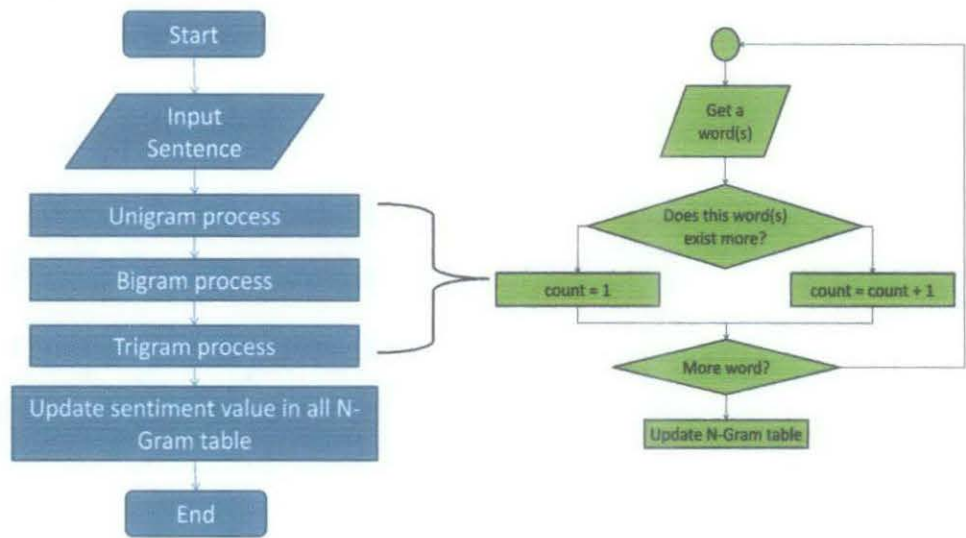


Figure 5 : Input process flow chart and sub flow chart where all N-Gram process will go through it.

The purposes of designing this process are to feed the N-Gram tables with English words and to continuously update the N-Gram tables. This data is very crucial because it will use in the analyzing process later on.

Summarize of input process:

- 1. Segment the word(s) for all N-Gram type. If Unigram, one word. If Bigram, two words. If Trigram, three words.
- 2. Count the occurrences of segmented words.
- 3. Update the N-Gram tables.

## 2.Analyzing Process

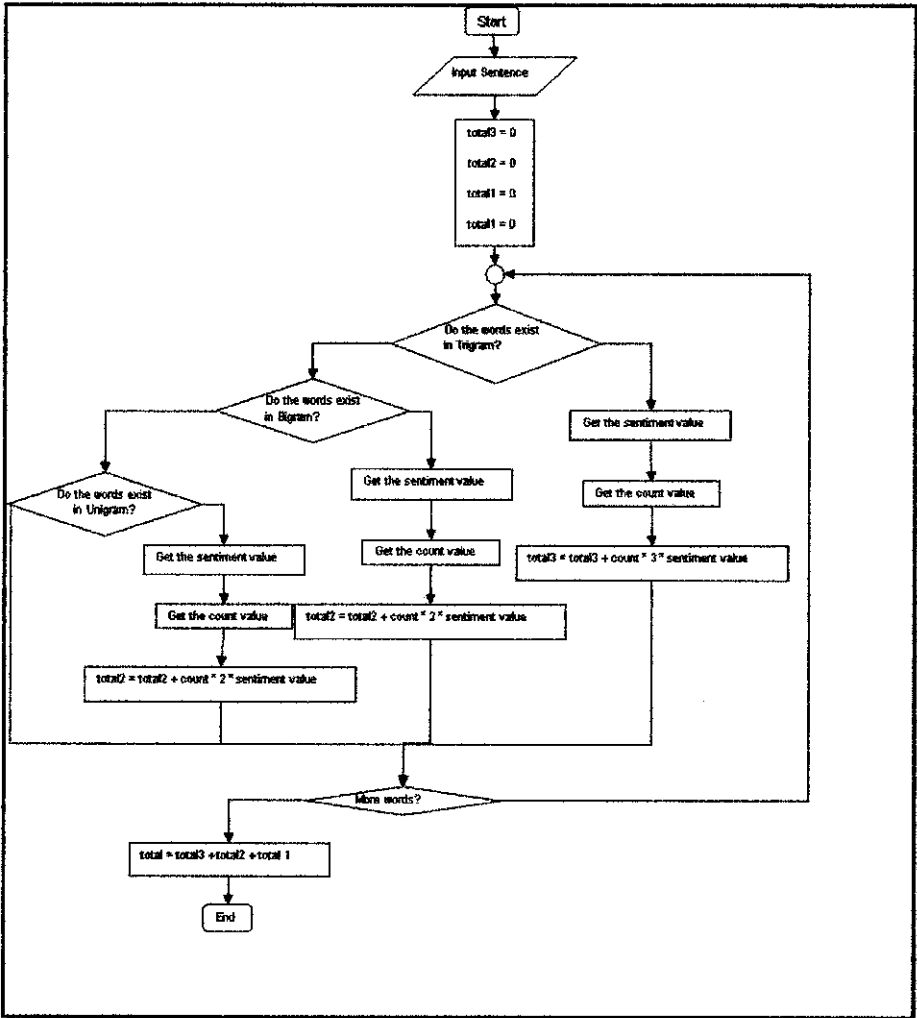


Figure 6 : Analyzing process flow chart

Different from input process, the analyzing process is started from the biggest N-Gram size which is Trigram. Then, it continues to Bigram and lastly Unigram. After that, the result will be sum up. If the total result is negative, the sentiment for the sentence is negative. If the total result is positive, the sentiment for the sentence is positive. The sentiment would be neutral if the result is 0.

Multiply the number of occurrence from N-Gram table with the N-Gram size. Then, multiply it with the sentiment value (-1=negative, 0=neutral or 1=positive). The analyzing process can be summarize in pseudocode below:

1. Get the input sentence
2. Capture first 3 words in the sentence.
3. If these words have in *Trigram* table, get the count and sentiment value.
4. Calculate the N-Gram score and N-Gram count for that word(s).

$$N\text{-Gram Score} = \text{Count} \times \text{Polarity} \times N\text{-Gram Size}$$

$$N\text{-Gram Count} = \text{Count} \times N\text{-Gram Size}$$

5. Repeat 2. until 4. by changing the words(3/2/1 word(s)) and N-Gram table(Trigram, Bigram and Unigram).
6. Repeat process 5. until there is no words that do not being calculated.
7. Calculate Total N-Gram Count and Total N-Gram Score.

$$\text{Total N-Gram Count} = \text{Sum of all N-Gram Count}$$

$$\text{Total N-Gram Score} = \text{Sum of all N-Gram Score}$$

8. Calculate the result in percentage format.

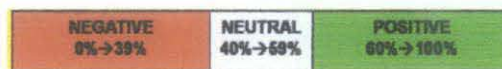
$$\text{Result}(\%) = \text{Total N-Gram Score} / (\text{Total N-Gram Count} \times 2) \times 100$$


Figure 7 : Range of result.

Lastly, the result will be determined whether it is positive, negative or neutral based on the score range above.



## CHAPTER 5: RESULT AND DISCUSSION

The author has already developing a working prototype that implement N-Gram model.

The features of this prototype include:

- Calculate Unigram, Bigram and Trigram word count in a sentence.
- Updating Unigram, Bigram and Trigram database table.
- Calculate sentiment of the sentence.

This is how the prototype GUI looks like:

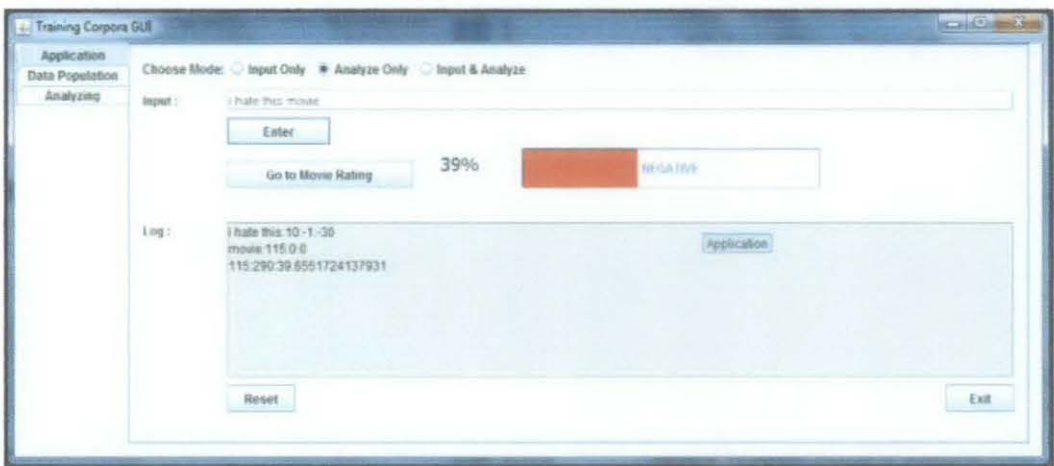


Figure 8 : GUI view from Application tab

This screen shot taken from Application's view where this is place to get user input sentence. User can write the sentence in input text area. Then, the user need to choose the mode (Input, Analyze, Input & Analyze). After the user hit enter, this prototype will calculate the unigram and bigram that occurs in the sentence. Then, the output will be display in output text area.



Mode:

- Input – the prototype will only input the new entered words by the user into the tables
- Analyze – the prototype will only analyze the sentence and give the result in the bar. The new entered word will not be inserted into the table.
- Input & Analyze – both process will occurs. First the N-gram table will be updated by the newly entered word and analyze back the word. Result (Positive, Negative or Neutral) will be produced in the bar.



The screenshot shows a software interface titled "Training Corpore GUI". On the left is a sidebar with "Application" and "Data Population" tabs, with "Data Population" selected. The main area contains three tables: "Unigram:", "Bigram:", and "Trigram:". Each table has columns for "no.", "word1", "word2" (or "word3" for Trigram), "count", and "polarity". Below the tables are "Query" and "Clean Data" buttons.

no.	word1	count	polarity
168	camera	1	0
169	can	9	0
170	candy	1	0
171	cannot	1	-1
172	cards	1	0
173	carrying	1	0
174	cast	2	0
175	cast	1	0
176	Cafe	1	0
177	cater	1	0
178	century	1	0
179	certain	1	0
180	CGI	1	0
181	CGI	1	0
182	chance	1	0
183	chapter	1	0
184	chara	3	0

no.	word1	word2	count	polarity
282	charact	who	1	0
283	charact	that	1	0
284	charact	in	1	0
285	childlike	but	1	0
286	Christ	seaso	1	0
287	cinema	that	1	0
288	city	objects	1	0
289	cold	up	1	0
290	colors	other	1	0
291	come	with	1	0
292	come	out	1	0
293	come	to	1	0
294	comic	relief	1	1
295	compa	Sam	1	0
296	compare	it	2	0
297	compe	but	1	0
298	concer	this	1	0

no.	word1	word2	word3	count	polarity
196	by	this	movie	2	0
197	by	ITSELF	as	2	0
198	can	not	be	1	-1
199	can	be	done	1	0
200	can	single	slaughter	1	0
201	can	gum	up	1	0
202	cardbo	charact	self-im	1	0
203	Cafe	Blanch	who	1	0
204	cater	for	his	1	0
205	chapter	of	a	1	0
206	character	that	doesn't	1	-1
207	character	develop	null	1	0
208	character	who	made	1	0
209	chopped	off	by	1	0
210	church	MUSIC	null	1	0
211	comes	to	an	1	0
212	comic	relief	as	1	0

Figure 9 : Data Population tab

In Data Population tab, the user can view the latest data that stored in the database table (Unigram and Bigram). By clicking Query button, all information will be displayed. Clear Data button is used for deleting all data in the database or in other word resetting the database table. Initially, the author intended to make a SQL Dashboard that can be used for any data manipulation process.

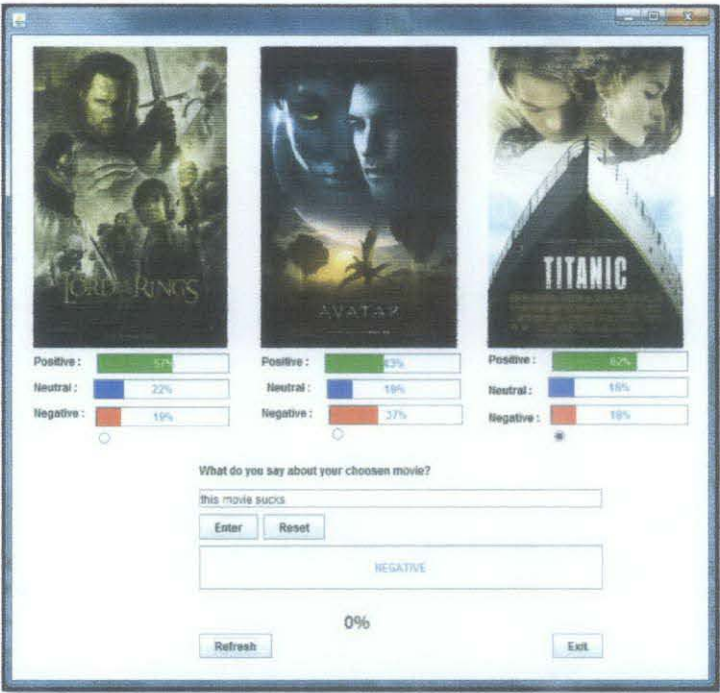


Figure 10 : Sample Application on Movie Rating

This sample application shows the statistical information that can be displayed after a number of user sentences being inserted into the prototype. The information is very useful for people who makes the movie or for a movie theater. They can make better business decision by using such information later on.

To test the accuracy of the prototype, a real sentence will be feed and analyze. The test sentence was from Metacritics.com. Here, internet user can make review for many types of products (i.e. games, movie, tv and music). The movie sample is The Lord of The Rings: Return of the King. After testing it, the result then populated in the table below.

Polarity	Test Size	Result	Accuracy(%)
Positive	5	4	80
Neutral	5	4	80
Negative	5	1	20

Figure 11 : Result and Accuracy Rate

The accuracy rate for Positive and Neutral sentiment is 80%. Whereas, for Negative sentiment the accuracy rate only 20%. This means that only 20% of the sentence for

Positive and Neutral sentence does not give correct result. For Negative sentiment, it has 80% sentence that does not give accurate result. This is because most of the words contains in the sentence is neutral words. That's how it affected the sentiment result. Most of the time, the result is affected by the N-Gram count in the N-Gram table. More count will give accurate result. So, the N-Gram table needs to continuously update with English words.

### N-Gram Application Class Diagram:



<b>NGramParser</b>
NGramParser();
-subjects:String -uL:Unigram[] -biword:String -bigramlist:Bigram[] -bigramfinal:Bigram[] -trigramlist:Trigram[] -trigramfinal:Trigram[]
+setSubjects(subjects:String):void +getSubjects():String +tokenizing(subjects:String):void +checkSimilar(word1:String, word2:String):boolean +compareUpdate(data:Database):void +printOutput():String +toArray(String paragraph):String[] +segmentation(sentence:String):void +TriSegmentation(sentence:String):void

<b>AnalyzeString</b>
AnalyzeString();
-word_count<String, Integer>:Map -word_polarity<String, Integer>:Map -sentiment:int -scorerange:int -realnumber:int -result:double -output:String
-setScoreRange(scorerange:int):void -getScoreRange():int -setSentiment(sentiment:int):void -getSentiment():int -setRealNumber(realnumber:int):void -getRealNumber():int -toArray(String sentences):String[] -setResult(result:double):void getResult():double -insertHashMap():void -setWordArray(sentence:String):void -getWordArray():String[] -calculateSentiment(word:String[], result:int, max:int):void -setOutput(word:String, polarity:int, score:int):void -getOutput():String

<b>Database</b>
Database();
-con:Connection -stmt:Statement -rs: ResultSet +uniList:Unigram +biList:Bigram
+connect():void +closeCon():void +executeSQL(statement:String):void +insert(word:String, count:int):void +insert(word1:String, word2:String, count:int):void +delete():void +retriveData():void +retriveData2():void +updateData(word:String, count:int):void +updateData(word1:String, word2:String, count:int):void

**N-Gram Database tables structure:**

Unigram	Bigram	Trigram
word	word1	word1
count	word2	word2
polarity	count	word3
	polarity	count
		polarity

## **CHAPTER 5: CONCLUSION AND RECOMMENDATION**

### **Conclusion:**

This paper was about two different topics that can work between each other. The topics are sentiment analysis and N-Gram algorithm. The purpose of sentiment analysis is to come out with result that shows on how human feels towards certain subject. In this project, the author used movie as his subject. This can achieve by implementing N-Gram algorithm.

In earlier part of this report, the author had discussed about sentiment analysis. The key concept in sentiment analysis was explained detailed in Literature Review. Generally, sentiment means human feelings, opinion or perception to something; it could be product, service or political issue. There are many researches that have been done in sentiment analysis. This area of natural language processing has increasingly become popular among businesses because they can monitor how customer feels about their product directly. Many applications have already being developed using various techniques in order to analyze human natural language. But, some of them do not give an accurate result. However, it can be improve by using computational linguistic methods like machine learning algorithm or N-Gram algorithm.

Next, the author also has discussed about N-Gram algorithm and how it works by using example. There are many usages of N-Gram algorithm. It can be use to categorize text/document and also used in text prediction.

Then, this paper also explains on how this research project being conducted in Methodology section. The author use two different approach respectively (prototyping method and experimental method). Software prototyping is highly related with experimental method because the author wills frequently testing his hypothesis on the prototype.

In Result and Discussion section, the author come out with sample of prototype design that can be use to demonstrate on how N-Gram algorithm works. This process

only can be done after the author has a concrete understanding on how to implement N-Gram algorithm and some probabilistic understanding.

The author hopes that this project can be a major contribution to the industry and helps business to improve their customer relationship.

### **Recommendation:**

#### **1. Good program design**

There are several things that can improve this prototype such as efficient Java class design. As shown earlier, the author designs the Java class for each N-Gram type (i.e. Unigram, Bigram, Trigram, etc). This may not be a good design. The author needs to create another class if he wants to implement new type of N-Gram.

#### **2. Crawling Program**

To make this prototype works more effectively and produce accurate result, the N-Gram databases need to be updated continuously from time to time with English article. A crawling program would be helpful to do this. This crawler program can find any new word through website and update the N-Gram database. If the author has the resources, he would be very happy to build this crawler program.

#### **3. Support Multilanguage**

Currently, this prototype only supports English language. In future development, the author wants to include other language like Bahasa Melayu. So, not only it can determine the sentiment value for English, it also can determine the sentiment value for other language. If the author wants to implement Bahasa Melayu in the prototype, he only needs to make modification on the N-Gram database tables (Unigram, Bigram and Trigram).

#### **4. Available on the Internet**

As mentioned in the beginning of this report, people always express their feelings on the Internet. So, there are vast of English words collection waiting to

be analyzes. If this kind of application is available on the Internet, it is accessible by everyone and people can use it. They can monitor the user comments on certain product. They also can know how many people positive, negative or neutral. This 'web application' can be like 'Twitter' but the different is that this application can determine the sentiment of a given sentence.

#### **5. Include Statistical Tools**

Business decision makers can make effective decision through graph, chart or table. Currently, there are no statistical tools included in the prototype. If graph included, businesses can monitor how their product sound in the heart of the customer. Then, he/she can decide what to do their marketing in order to attract more customers and increase positive feedback from them.

#### **6. Support Bigger N-Gram size**

In this prototype, it can only support Unigram, Bigram and Trigram. As shown earlier, if the N-Gram size is increase, the more accurate result will be produced. So, if this application can apply more N-Gram like Quadgram (4-gram), 5-gram and so on, the result produced will be accurate.



## REFERENCES

1. Sentiment <<http://en.wiktionary.org/wiki/sentiment>>
2. Sentiment Analysis <[http://en.wikipedia.org/wiki/Sentiment\\_analysis](http://en.wikipedia.org/wiki/Sentiment_analysis)>
3. Bing Liu, Department of Computer Science, University of Illinois at Chicago, 2010, "*Sentiment Analysis and Subjectivity*"
4. Bo Pang, Yahoo! Research, Lillian Lee, Computer Science Department, Cornell University, 2008, "*Opinion Mining and Sentiment Analysis*"
5. Tweetfeel FAQ <<http://www.tweetfeel.com/faq.php>>
6. Twitrratr About <<http://twitrratr.com/about>>
7. Twendz About <<http://twendz.waggeneredstrom.com/about-twendz.aspx>>
8. N-gram <<http://en.wikipedia.org/wiki/N-gram>>
9. Peter Nather, Faculty of Mathematics, Physics and Informatics, Institute of Informatics, 2005, "*N-gram based Text Categorization*"
10. "The English Language: 900,000 Words, and Counting" <<http://www.npr.org/templates/story/story.php?storyId=5182871>>
11. Software Prototyping <[http://en.wikipedia.org/wiki/Software\\_prototyping](http://en.wikipedia.org/wiki/Software_prototyping)>
12. Steps of the Scientific Method <[http://www.sciencebuddies.org/science-fair-projects/project\\_scientific\\_method.shtml](http://www.sciencebuddies.org/science-fair-projects/project_scientific_method.shtml)>
13. M. Zubair Shafiq, Syed Ali Khayam and Muddassar Farooq, "*Embedded Malware Detection using Markov n-grams*"

