

NATURAL LANGUAGE PROCESSING TECHNIQUE FOR SENTIMENT

ANALYSIS



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

NATURAL LANGUAGE PROCESSING TECHNIQUE FOR SENTIMENT

ANALYSIS



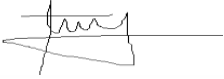
Bachelor of Computer Science (Artificial Intelligent)


FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TEKNIKAL MALAYSIA MELAKA 2016

DECLARATION

I hereby declare that this project report entitled
NATURAL LANGUAGE PROCESSING FOR SENTIMENT ANALYSIS
is written by me and is my own effort and that no part has been plagiarized
without citations.

STUDENT :  _____ Date: 15-08-2016
(TAN ZHEN YEE)



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

I hereby declare that I have read this project report and found this project report is
sufficient in term of the scope and quality for the award of Bachelor of Computer
Science (Artificial Intelligent) with Honours

SUPERVISOR : *lieza* _____ Date: 15-08-2016
(DR HALIZAH BINTI BASIRON)

DEDICATION

I would like to dedicate my final year project report to my beloved supervisor, family, and friends for being supporting and help me during this final year project. Especially my supervisor Dr. Halizah binti Basiron, who was always guided me and help me to complete up this project. Thanks for all support.



ACKNOWLEDGEMENTS

First of all, I would like to take this opportunity to express my sincere gratitude to my supervisor Dr. Halizah binti Basiron who had given me full support, patience, motivation, guide and encouragement along the progress of final year project. This project will not be success without her persistently assistant.

Besides that, I would like to thank my senior Nurul Fathiyah binti Shamsudin who are willing to take her own time to teach me how to use Python and help me solve error of coding. Therefore, I would like to thank for her enthusiasm and immense knowledge in this few months.

Lastly, I would like to take this chance to thanks my beloved family members who are always supporting me when I feel disapointed due to some problems happen in final year project.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ABSTRACT

Internet is getting useful in 21th century, so there are lots of user in the Internet. Most of the users will leave their comments in the net to express their emotion through words and emoticon in the social media like Twitter. Twitter is a kind of social media that are famous among Malaysians, therefore sentiment analysis research is carry out for some of the Malay version of tweet. Sentiment analysis is a kind of process for determining a particular words in to positive, neutral, and negative where it is widely used for deriving the opinion from social media such as Twitter. Most of the sentiment analysis is done for the use in the marketing area to know the review of customer. Therefore, the aim of doing sentiment analysis research is to differentiate the comment of tweet into three categories which are positive, neutral and negative. The Natural Language Processing (NLP) is used to identify the sentiment. The technique used is identifying sentiment by calculating the numbers of combination of word categories such as noun, adjective, adverb and verb. The data pre-processing is needed before using the NLP technique. After the experiment, the accuracy of the technique will be measured.

ABSTRAK

Internet semakin berguna dalam abad ke-21, jadi terdapat banyak pengguna di Internet. Kebanyakan pengguna akan meninggalkan komen mereka di internet untuk menyatakan emosi mereka melalui kata-kata dan emoticon dalam media sosial seperti Twitter. Twitter adalah sejenis media sosial yang terkenal di kalangan rakyat Malaysia, oleh itu penyelidikan analisis sentimen dijalankan untuk Melayu tweet. Analisis sentimen adalah sejenis proses untuk menentukan fakta yang tertentu kepada positif, neutral, dan negatif, ia digunakan secara meluas untuk memperoleh pendapat dari media sosial seperti Twitter. Kebanyakan analisis sentimen dilakukan untuk kegunaan di kawasan pemasaran untuk mengetahui kajian pelanggan. Oleh itu, matlamat untuk melakukan kajian analisis sentimen adalah untuk membezakan yang menurut tweet dalam tiga kategori iaitu positif, neutral dan negatif. The Natural Language Processing (NLP) digunakan untuk mengenal pasti sentimen. Teknik yang digunakan ialah mengenal pasti sentimen dengan mengira bilangan kombinasi kategori perkataan seperti kata nama, kata sifat, kata keterangan dan kata kerja. Data yang pra-proses yang diperlukan sebelum menggunakan teknik NLP. Selepas eksperimen, ketepatan teknik akan diukur.

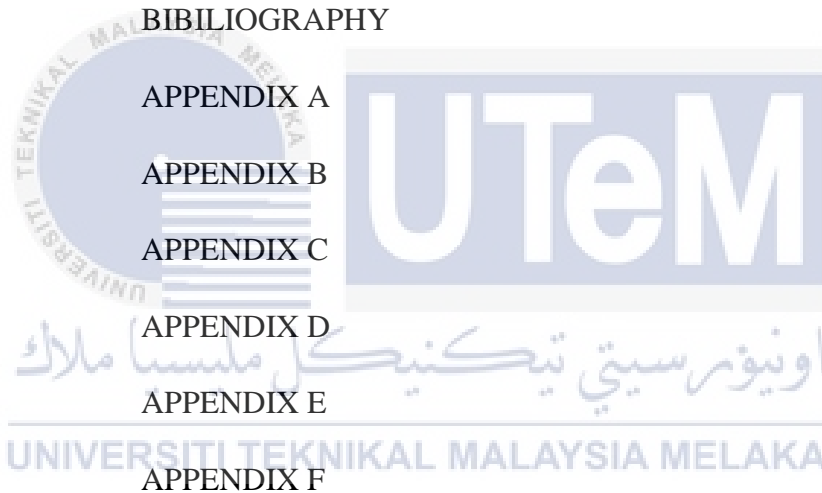
TABLE OF CONTENTS

CHAPTER	SUBJECT	PAGE
	DECLARATION	i
	DEDICATION	ii
	ACKNOWLEDGEMENT	iii
	ABSTRACT	iv
	ABSTRAK	v
	TABLE OF CONTENTS	vi
	LIST OF TABLES	x
	LIST OF FIGURES	xi
CHAPTER I	INTRODUCTION	
	1.1 Introduction	1
	1.2 Problem Statement	2
	1.3 Objective	2
	1.4 Scope	3
	1.5 Project Significance	3
	1.6 Expected Output	3

1.7 Conclusion	4
CHAPTER II	LITERATURE REVIEW
2.1 Introduction	5
2.2 Facts and Findings	5
2.2.1 Text Processing	6
2.2.2 Existing System	6
2.2.3 Technique	13
2.2.3.1 Machine Learning	13
2.2.3.1.1 Support Vector Machine (SVM)	13
2.2.3.1.2 Naive Bayes	14
2.2.3.1.3 Maximum Entropy	15
2.2.3.2 Natural Language Processing (Lexicon-based Approach)	16
2.2.3.2.1 Dictionary-based	16
2.2.3.2.2 Corpus-based	16
2.6 Conclusion	17
CHAPTER III	METHODOLOGY
3.1 Introduction	18
3.2 Problem Analysis	20
3.3 Requirement Analysis	20

3.3.1 Software Requirement	20
3.3.2 Hardware Requirement	21
3.4 Data Requirement	22
3.4.1 Pre-processing Data	22
3.4.2 Building Word Dictionary	23
3.4.3 Sentiment Tag	23
3.4.4 Calculate Sentiment Value	23
3.5 Project Schedule and Milestones	24
3.5 Conclusion	27
CHAPTER IV SENTIMENT EXTRACTION TECHNIQUE	
4.1 Introduction	28
4.2 Data Pre-processing	28
4.3 Dictionary-based Approach	31
4.4 Tagging Words	31
4.5 Extracting Sentiment from Tweet comments	32
4.6 Conclusion	33
CHAPTER V EXPERIMENTAL RESULT AND ANALYSIS	
5.1 Introduction	34
5.2 Experiment Result	35
5.3 Discussion	41

5.4 Conclusion	42
CHAPTER VI PROJECT CONCLUSION	
6.1 Introduction	43
6.2 Proposition for Improvement	43
6.3 Project Contribution	44
6.4 Conclusion	44
REFERENCES	45
BIBLIOGRAPHY	46
APPENDIX A	47
APPENDIX B	49
APPENDIX C	50
APPENDIX D	51
APPENDIX E	52
APPENDIX F	53



LIST OF TABLES

TABLE	TITLE	PAGE
2.1	Summary of Literature Review	9
3.1	Software Requirements	20
3.2	Hardware Requirements	21
3.3	Gantt Chart of Project Activities PSM1	25
3.4	Milestones and Dates PSM1	25
3.5	Gantt Chart of Project Activities PSM2	26
3.6	Milestones and Dates PSM2	26
4.1	Process of data pre-processing	29
5.1	Category of words	35
5.2	Summarize of experiment	40

LIST OF FIGURES

FIGURE	TITLE	PAGE
2.1	SVM in classification problem	14
2.2	Maximum entropy in classification problem	15
3.1	Waterfall model	19
3.2	Flow of Sentiment Analysis for Tweets	22
3.3	Flow chart of project activity	24
4.1	Sample of Malay tweets data with sentiment tag	31
4.2	Sample of MalaySentiWordNet with words category tag of adjective	33
5.1	Result of ar_addnv	36
5.2	Result of ar_addvn	37
5.3	Result of ar	37
5.4	Result of arn	38
5.5	Result of arv	39
5.6	Result of nv	39
5.7	Result of arnv	40
5.8	Accuracy of Sentiment analysis based on different combination of category of words	42

CHAPTER I



1.1. Introduction

Sentiment analysis is the use of natural language processing and text analysis to identify the information of resources by determining a particular words statement is either positive, neutral, and negative polarity. Sentiment analysis is important for judging a new item or product and differentiate whether the comment more to positive or neural or negative feature. For example, a company need customers' feed back on their product in order to know whether their product is marketable or not.

This project focuses on sentiment analysis using NLP on Malay tweet data. For the NLP technique we need to pre-process the Malay version Tweet comment

before tagging and compare the output using Malay word database after that it will give score on that particular comment to determine whether it is positive or negative. This is because Twitter is a kind of social media that are famous among Malaysian, Malaysian like to express their emotion and comment at this social media. Based on this Twitter have become a place where people will collect and take as determination from the product review or other things. Sentiment Analysis has four types of categories which are keyword spotting, lexical affinity, statistical methods and concept-level techniques (Poria, S., Cambria, E., Winterstein, G. and Huang, G.B., 2014). The category of sentiment analysis we use in this project is lexical affinity. We will identify the Malay word by using Malay word database to ensure the accuracy.

1.2. Problem statements

In this digital world, computer can work faster than human being because human judgement may take times in process of differentiate something. Computer are work in automated form which mean that it is fast, accurate, efficient and less error. Humans have feeling and sometimes are unfair on the judgement, while computer do not have any feeling it just follow the instruction that set by us. Besides that most of current sentiment analysis research are doing their research for English version tweet but not much doing on Malay tweet. Therefore, in our project will focus on doing Sentiment Analysis on Malay tweets.

1.3. Objectives

- 1.To build a corpus of Malay words based on Malay tweets.
2. To analyse the word either positive, neutral or negative.
3. To implement lexicon based techniques on Malay tweets.

4. To compare results among lexical based techniques.

1.4. Scope

The scope for this project are installing Python 2.7.9, Numpy 1.6.1, Toolkit for coding need and source of Malay tweets, . For the data used for analysis are Twitter in Malay Language, which are new try for us. In this project will be focus on lexical-based which means the combination of words categories such as adjective, adverb, verb and noun, but does not focus on the syntax of sentence.

1.5. Project Significance

Sentiment Analysis in Twitter can detect the comment into negative, positive and neutral. Most of the comment for product will take as determination for the product review. This project may benefit the company because due to the comment the company will know whether people like their product or not. From the review that analyze the company can change the product that fulfil customer requests, so that company can sell their as many as possible.

1.6. Expected Output

Expected output of doing sentiment analysis research is to differentiate the comment of tweet in to three categories which are positive, neutral and negative. Our experiment will focus on adjective, adverb, noun and verbs in the comment to give score for the comment to classify it into positive, neutral or negative comment by using Natural Language Processing (NLP) technique and use Malay Word

Dictionary that have been build by our own to compare the word and give score on each adjective. After finishing the experiment we would measure the accuracy of the technique.

1.7. Conclusion

As a conclusion, this chapter has explained about the project on Sentiment Analysis that we developed. In the following chapter, the literature review will be further discussed.



CHAPTER II



2.1. Introduction

This chapter will explain about literature review conducted for sentiment analysis. The main purpose of this chapter is to summarize all the research found which apply similar or different technique for sentiment analysis. Therefore, literature review is very important because we can get a better result by reviewing all the experimented result from each research and use the experience from them to improve our own research.

2.2. Facts and findings

2.2.1. Text Processing

In natural language text, most of the information is implicit and when watch on it separately the meaning maybe ambiguous Ralph, G., & Principal, I., n.d. . Therefore, text processing are used for processing sentences in natural languages such as Malay, English, Mandarin, but not used for programming languages like Java or Php.

Sentiment analysis is a kind of text processing and text analysis to identify the information of resources into the category of positive, negative or neutral. Therefore, sentiment analysis is used to examine the feeling of people's comment in a particular post. In the previous research in sentiment analysis in (Pang, B., Lee, L. and Vaithyanathan, S., 2002), they have analyzed on the performance of different classifiers on movie review. Due to the work of Pang, B., Lee, L. and Vaithyanathan, S., 2002, we can use the same technique that provided in their paper in analyze Tweets.

Sentiment Analysis have four type of categories which are keyword spotting, lexical affinity, statistical method and concept-level technique. The category of sentiment analysis we use in this project are lexical affinity, we will identify the Malay word by using Malay word database to ensure the accuracy. Sentiment analysis is important for judging a new item or product and differentiate whether the comment more to positive, neutral or negative feature, because human will be more understand based on analysis we perform using Natural Language Processing (NLP) technique. For the NLP technique we need to pre-process the Malay version comment before tagging and compare the output using Malay word database after that it will give score on that particular comment to determine whether it is positive, neutral or negative.

2.2.2. Existing System

Most of the sentiment analysis is done for the use in the marketing area to know the review of customers. There are many sentiment analysis related research done, by using machine learning algorithm like Support Vector Machine(SVM),

Naive Bayes and Maximum entropy classification , B., Lee, L. and Vaithyanathan, S., 2002. All the technique as mentioned at above sentences are used to perform classification task in classify movie review based on the tagging in each of adjective word into positive and negative. The data set used by this research paper was Internet Movie Database (IMDb) with less than 20 review per author and per sentiment category. Therefore, there are resulting 752 negative review and 1301 positive review. The results show that Naive Bayes have the worst result compared to SVM.

In the paper of Khong, W.H., Soon, L.K. and Goh, H.N., 2015 , they have tried on machine learning technique SVM, Naive Bayes, k-Nearest Neighbor and also Natural Language Processing (NLP) by using data from twitter comment and extract 3424 tweets from it to do the experiment. They found out that SVM provide higher accuracy of sentiment analysis because it can handle multiple continuous and categorical compared with other machine learning algorithm. Besides that when they try on Natural Language Processing (NLP), they found that it have a low accuracy compare with all machine learning technique.

In the paper of Agarwal, A., Xie, B., Vovsha, I., Rambow, O. and Passonneau, R., 2011 they train 11,875 manually annotated tweets from a commercial source by using SVM technique then compare with sentiment analysis on micro-blog data. At the end they found that sentiment analysis for Twitter data do not have much different from sentiment analysis for other genres.

There are some of the researcher doing sentiment analysis by using several dataset instead of only taken one kind of dataset. In paper of Xia, R., Zong, C. and Li, S, 2011 they have use Naive Bayes, Maximum entropy classification and SVM for training data of Cornell movie review, Multi-Domain Sentiment Dataset, and product reviews taken from Amazon.com, and it shows that SVM get a better result compare with Naive Bayes and Maximum entropy.

In the paper of Somprasertsri, G. and Lalitrojwong, P., 2010 they have tried on Maximum entropy for doing sentiment analysis in digital camera review which taken form Amazon.com. They randomly taken 1250 sentences from the review as their data and they found that Maximum entropy show an good result on it.

For paper of Jian, Z., Chen, X. and Wang, H.S., 2010 they propose to use theory of Artificial Neural Network for doing sentiment classification by using 1200 comments from movie review. They have proved that by using back propagation

method it shown almost 87% of accuracy, and the advantage of this method is it can automatic extracts the sentimental features without sentimental word dictionary.

Twitter Sentiment Classification done by Go, A., Bhayani, R. and Huang, L., 2009 have use the technique of Baseline Naive Bayes, Maximum Entropy and Support Vector Machines. They use tweets which taken from April 6, 2009 to June 25, 2009. After going through the process it shows that the accuracy of SVM higher than the rest of technique.

In the paper of Kouloumpis, E., Wilson, T. and Moore, J.D., 2011 they have tried on sentiment lexicon technique and use three different corpora of Twitter messages which are hash tagged data set, emoticon data set and iSieve data set. After, they have finish their work they found that sentiment lexicon were useful in conjunction with microblogging features.

This paper used SVM as a technique for sentiment analysis approach by using movie review as data. They have taken 1380 comment form imdb.com movie reviews as the training dataset, and after running they get result which are higher than 80% in the paper of Mullen, T. and Collier, N., 2004

Lexicon approach is a kind of technique for sentiment analysis. It need to calculate the polarity then continue with calculate the subject score, if the score are negative mean it is negative post, while when the score are positive it is a positive post. The data used for this researcher are taken from blog posts form the paper of Godbole, N., Srinivasaiah, M. and Skiena, S., 2007. The summary of literature review is outlined in Table 2.1 Literature summarize.

Table 2.1 A summary of literature review

NUMBER	YEAR/ AUTHOR(S)	TECHNIQUES	DATA USED	RESULT	TRAINING DATA SET	FUTURE WORK
1.	2002 B. Pang, L. Lee, H. Rd, and S. Jose	SVM, Naive Bayes, and Maximum entropy classification	Internet Movie Database (IMDb)	Naive Bayes have a worst result compared to SVM	20 reviews per author per sentiment category	They will be focus on identification of features indicating whether sentences are on-topic
2.	2015 Khong, W., Soon, L., & Goh, H	SVM , Naive Bayes, k-Nearest Neighbour, and Nature Language Processing	Twitter comment	Natural Language Processing (NLP) low accuracy compare with machine learning technique	3424 tweets	They will work on extract and identify the words that only extract the sentence with the subject.
3.	2011. Agarwal, A., Xie, B., Vovsha, I., Rambow, O.	SVM	Twitter comment	Sentiment analysis for Twitter data do not have much different from sentiment	11,875 manually annotated tweets from a commercial	They will be focus on exploring linguistic analysis

	and Passonneau, R			analysis for other genres	source	
4.	2011 Rui, X., Chengqing, Z., & Shoushan, L	Naive Bayes, Maximum entropy, SVM	Cornell movie review, Multi-Domain Sentiment Dataset, and product reviews taken from Amazon.com	SVM show higher accuracy compare with Naive Bayes, and Maximum entropy		They will work on feature selection for syntactic relations for future
5.	2010 Somprasertsri, G., & Lalitrojwong, P	Maximum entropy	Digital camera review form Amazon.com	It effectiveness on the proposed approaches	1250 sentences from customer review	In the future they hope to investigate self-learning methods for classification that can reduce the amount of labelled data required to produce highly accurate

						results.
6.	2010 Jian, Z., Chen, X., & Han-shi, W	Back propagation	Cornell movie review	Accuracy are close to 87%.	1 200 movie comment	
7.	Go, A., Bhayani, R., & Huang, L.	Baseline, Naive Bayes, Maximum Entropy, Support Vector Machines	Twitter comment	Accuracy for Naive Bayes are 81.0%, Maximum Entropy have 80.4% and SVM have 82.9%	Tweets taken from April 6, 2009 to June 25, 2009	They hope that in the future can get a more accurate result.
8.	Kouloumpis, E., Wilson, T., & Moore, J.	Sentiment lexicon	Twitter comment	Sentiment lexicon were useful in conjunction with microblogging features	Hashtagged data set, Emoticon data set, iSieve data set	
9.	2004 Mullen, T., & Collier, N	SVM	Movie review	Accuracy above 80%	1380 comment form imdb.com movie reviews	

10	2007 Godbole, N., Srinivasaiah, M., & Skiena	Lexical approach	Blog posts			They hope to get a degree for the sentiment indices to predict future changes in popularity or market behaviour.
----	---	---------------------	------------	--	--	---



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2.2.3. Techniques

Sentiment Analysis technique can be divided into two types of approach: Machine Learning approach and Lexicon-based approach.

2.2.3.1 Machine Learning

Machine Learning Approach (ML) for the text classification can be divided into supervised learning and unsupervised learning methods, but most frequently ML technique used by Sentiment Analysis are supervised learning method like Support Vector Machine (SVM), Naive Bayes, and Maximum Entropy . The supervised methods will used labelled training dataset for getting the result.

2.2.3.1.1 Support Vector Machine (SVM)

Support vector machines (SVM) are supervised learning models that can be used for statistical classification and regression analysis. SVM is the best text classification method (Rui Xia, 2011). SVM seeks a decision surface of the training data points and then separated it linearly with a hyperplane into two classes and makes decisions based on the support vectors that are selected as the only effective elements in the training set. Therefore, SVM are used as a sentiment polarity classify because we can use it after identifies and extract the comment of Twitter in the opinion associated In the most of the research paper SVM show the best result compare with other machine learning methods in sentiment classification. Figure 2.1 have shown application of SVM technique in classification problem.

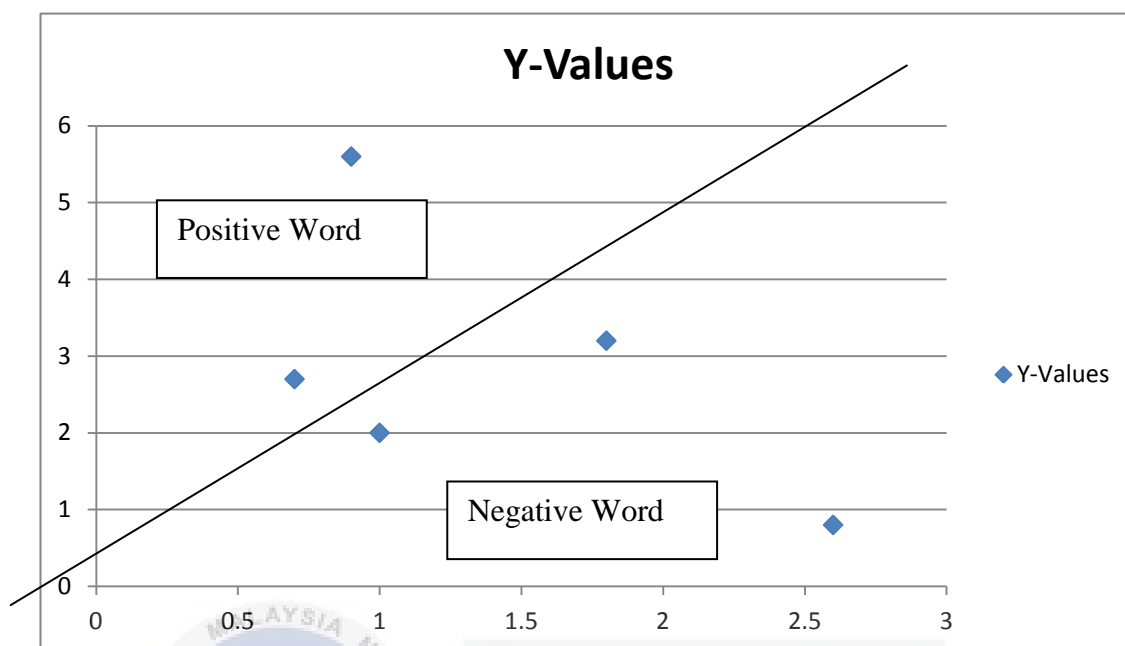


Figure 2.1 SVM in classification problem.

2.2.3.2.2 Naive Bayes

Naive Bayes is a simple but effective classification algorithm and it is widely used algorithm for document classification (Medhat, W et. al.). Naive Bayes is used to estimate the probabilities of categories given a test document by using the joint probabilities of words and categories. Naive Bayes are using conditional probability in the Bayes Theorem. For an example by using equation of Naive Bayes $P(y|x) = \frac{P(y)P(x|y)}{P(x)}$, let's $P(y)$ is represented as the probability for positive tweets, $P(x)$ is represented as the probability for tweet, $P(x|y)$ is represented as the probability of x , the positive tweet given y , the tweet. $P(x)$ are used in selecting $P(x|y)$. $P(x)$ is a constant so that it is not important in the calculation, therefore, the calculation part will be focus on $P(x|y)$ and $P(y)$. To calculate $P(x|y)$, it need a training set of tweets have been classified into the three categories, where this will give a basis from which to compute the probability that a tweet will fall into a specific class.

2.2.3.2.3 Maximum Entropy

Maximum entropy classification is an alternative technique that is effective when used in natural language processing applications. Maximum entropy work by choosing the least biased distribution, which maximizes uncertainty in the distribution subject to given constraints. Maximum entropy are better than Naive Bayes although both of them using conditional probability, because it does not make assumption between features. Formula of Maximum entropy $P_{ME}(c|d) = \frac{1}{Z(d)} \exp(\sum_i \lambda_{i,c} F_{i,c}(d, c))$ where $Z(d)$ is a normalization function, $\lambda_{i,c}$ is a weight vector, $F_{i,c}$ is a feature, c is the class and d is the tweet. The weight of vector will decide the significance of a feature in classification. The higher the weight the stronger the indicator for the class. Figure 2.2 have shown application of Maximum entropy technique in classification problem.

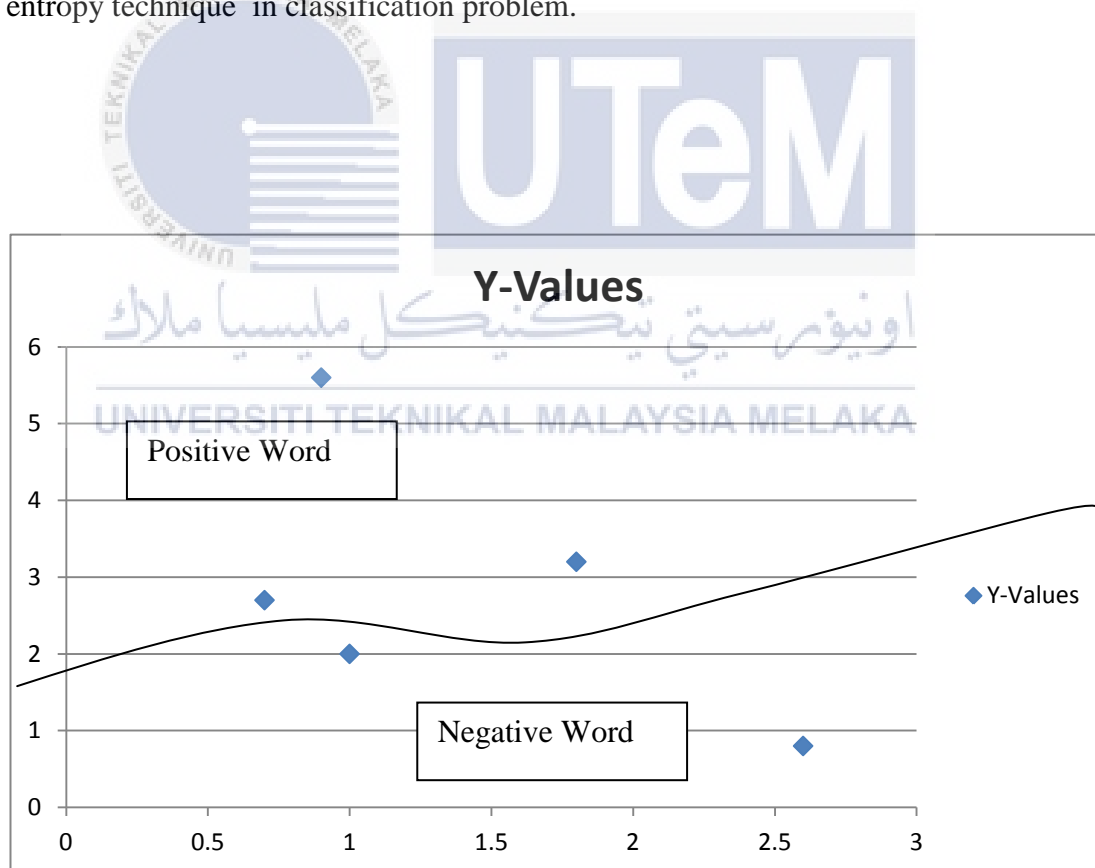


Figure 2.2 Maximum entropy in classification problem.

2.2.3.2 Natural Language Processing (Lexicon-based Approach)

The Lexicon-based approach are functioning by using the opinion lexicon to analyze the text. This approach can be divided into two methods which dictionary-based approach and corpus based technique.

2.2.3.2.1 Dictionary-based

Dictionary-based approach works by searching the similarity word on the dictionary. First step for dictionary-based approach is applied tagging methods on the Malay Tweets, then send the result after tagging for Word Sense Disambiguation (WSD) where in this place it will identify the Malay word by using MalayWordNet. After that, the word level sentiment analysis identification is performed by using SentimentWordNet (Baccianella et al., 2008) . After that it will undergo judgment on the word for each word. When all the words in the comment have finished tag the system will sum up all the score get by the comment to differentiate into positive or negative. Therefore, dictionary-based approach are using a rule based approach to catch a particular word and user attitude identification in advertising keyword extraction.

2.2.3.2.1 Corpus-based

Corpus-based approach are used to finds other opinion words in a large corpus to help in finding opinion words with context specific orientations (Medhat, W et. al.,) and it is performed by using statistical approach or semantic approach. For the statistical approach it need to find the seed opinion like using adjective in corpus to derive posterior polarities. The polarity of a word can be differentiate by looking on the frequency of the word occur in the corpus that created. Therefore if the word more frequently appear as positive, it will classify under positive polar, while if the word more frequently appear as negative, it will classify under negative polar, if the

frequency of both positive and negative appear are same then it will consider as neutral polar. For the semantic approach, it will give sentiment values directly to semantically close word. Therefore a WordNet is needed to keep a list of words with its sentiment value and synonyms where it will the words that will be used are verbs, nouns, and adjective. In this approach it need a deep describe for the detail of subjectivity relations in between the words in a sentence to expresses and separate polarity for each of the words. Therefore the words will be differentiate into positive, negative or neutral based on its subjectivity relations.

2.6. Conclusion

This chapter is a literature review that summarize all the research for the researcher for this project. Based on all the research it shows that this project can be improve. Other than that, methodology also have some briefs explain on the flow of the project, to ensure the project can move smoothly.

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER III



3.1. Introduction

We have chosen Waterfall Model as methodology to use in this project. This is because it is easy to manage due to the rigidity of the model which each phase has specific deliverable and a review process. Which mean that we can refer back to stages that we go through, once we found that there are something wrong in the current state. Figure 3.1 have shown the flow of Water Fall model used in this project.

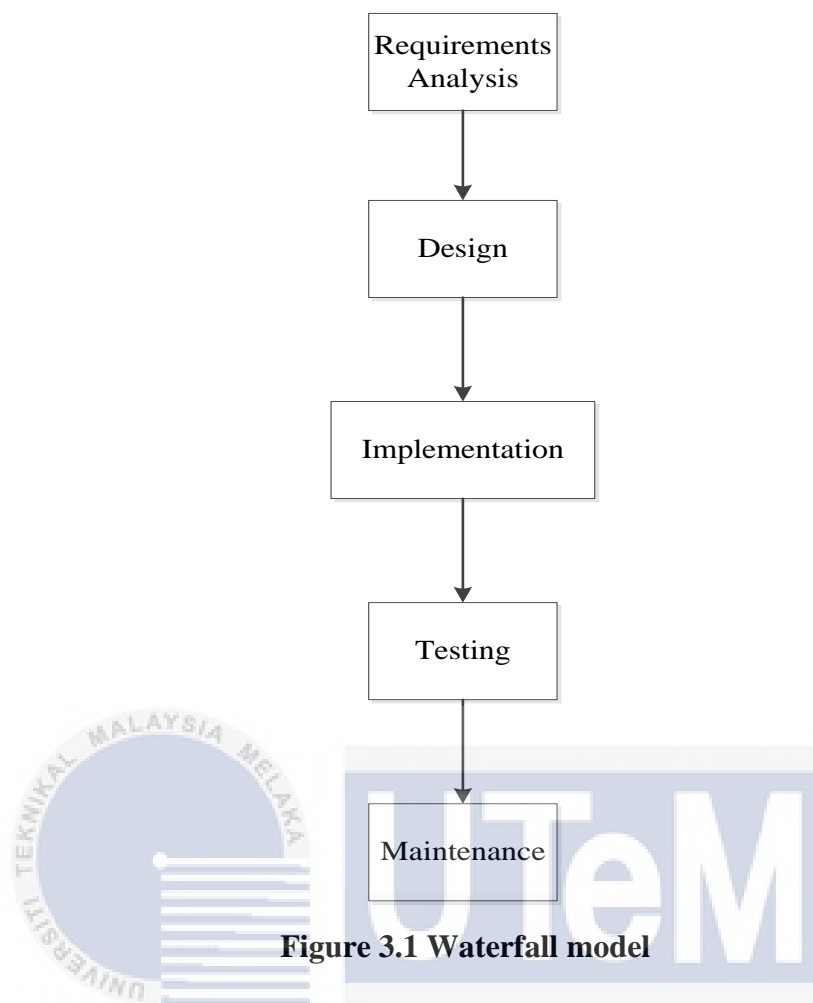


Figure 3.1 Waterfall model

In the requirement analysis phase, some research analysis regarding Sentiment Analysis will be done for collecting the idea and method for doing this project. Besides that, data analysis will be done for collecting the data from the social media of Twitter.

There will be several steps done in the design phase, where the steps are, drawing a flow for the project, installing software, and data pre-processing. The software that is used are Python of version 3.5.1, Numpy of version 1.9.2, and NLTK.

The next step of the waterfall model is the implementation phase, in this phase coding will be written for loading data, exporting data, comparing data and calculating sentiment value. Word tokenization will be done in this phase by loading the data into Python. After the word tokenization has been done, it will continue to do for the sentiment value tag that is the most important step for the calculation part.

For the testing phase, the coding will be run many times to ensure that the code is running stably with good performance.

Last but not least, in the maintenance phase are used to ensure that the project can be run correctly and with a stable performance.

3.2. Problem Analysis

Nowadays comments are very important for improving some products or quality of work. There will be some of the business man taking the review of product form Twitter to analyze the quality of the product. Therefore, there will be a need for Sentiment Analysis to classify the comment in Twitter into positive and negative.

In this digital world, computer can work faster than human being because human judgment may take times in process of differentiate some times and computer are considered as automated which mean that it is fast, accurate, efficient and less error. Everyone have feeling and sometimes are unfair on the judgment, while computer do not have any feeling it just follow the instruction that set by us. Besides that most of current sentiment analysis research are doing their research for English version Tweet but not much doing on Malay Tweet. Therefore, in our project will focus on doing Sentiment Analysis regarding Malay Tweet.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

3.3. Requirement Analysis

3.3.1. Software Requirement

The table below show all the software requirements needed in this project.

Table 3.1 Software Requirements

Software required	Description
Microsoft Word	For editing documentation.

Microsoft Visio	For creating flow chart.
Microsoft Excel	For creating and store data
Microsoft Windows 8	Personal computer operating system
Google Chrome	Web browser for internet seaching.
Python 3.5.1	Software to develop system
Matplotlib	A tool for creating graph that support Python program
Numpy 1.9.2	Software that support Python program
Nature Language Toolkit	A tool for building Python program
Acrobat Reader DC	To read any documentation of PDF format

3.3.2. Hardware Requirement

The table below show all the software requirements needed in this project.

Table 3.2 Hardware Requirements

Hardware required	Description
4GB installed memory (RAM)	To ensure computer can run smoothly.
Minimum 100GB of storage	To ensure there is enough space to install all the software needed.
Keyboard	To control and give command to computer
Mouse	To control computer
Network card	Enable to connect to internet

3.4. Data Requirement

The figure below show the flow of Sentiment Analysis for tweets, where the first step are take a part of tweets from tweets data, continue with pre-processing data, building Word Dictionary, Sentiment Tag and the last step are calculate Sentiment value.

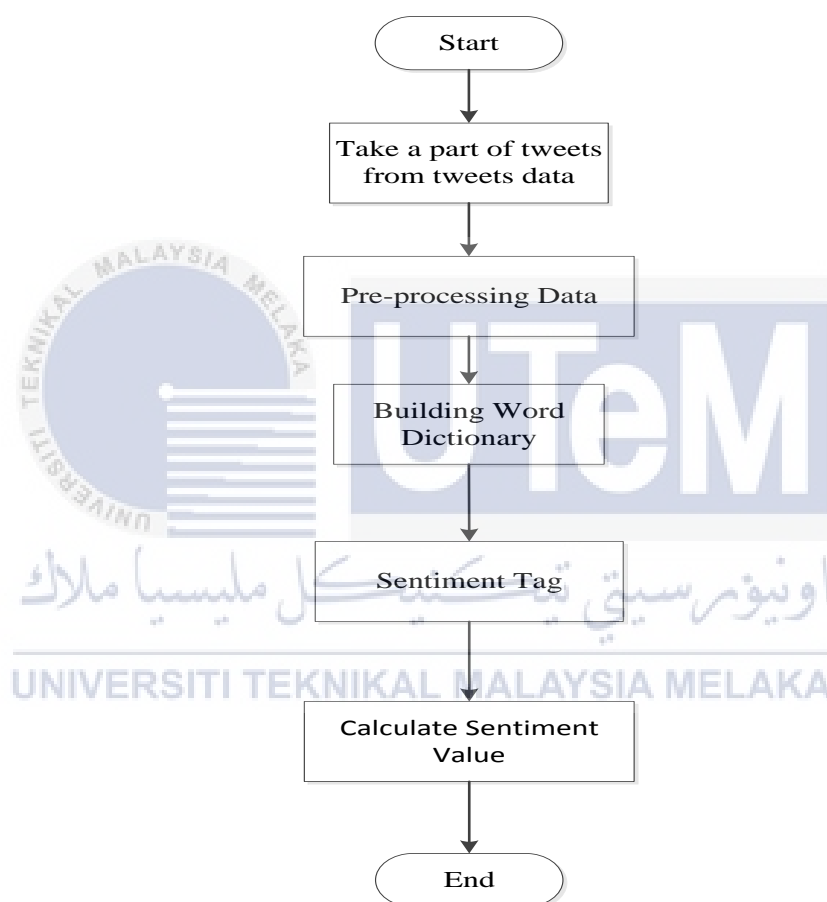


Figure 3.2 Flow of Sentiment Analysis for Tweets.

3.4.1. Pre-processing Data

The original data of twitter contain lots of noise that may affect result for sentiment analysis like hash-tag, internet link, short form of Malay words, and all kind of

emoticons that provided by twitter. To remove all the noises we will use manually method to clean up the tweet data.

3.4.2. Building Word Dictionary

After getting the clean data from section 3.4.1 we can start to split all the text into words, in this step will remove the word that have repeated, make sure each word appear only one time. Besides, the words with noun and pronoun will be removed and just extracting the adjective and verb words. The word that left will be saved as CSV file and it is the Malay Dictionary that created manually.

3.4.3. Sentiment Tag

For the data that have done in section 3.4.2, the data that collected can be tagged for its sentiment value for each of the word. For the negative word will be given a value of "-1", neutral word will be assign as "0" and a positive word will be given a value of "1". The words with their sentiment value are save as MalaySentiWordNet. Therefore MalaySentiWordNet done in this process.

3.4.4. Calculating Sentiment Value

In this step we will use the data set that have been clean in 3.4.1, to compare with the MalaySentimentWordNet that done in 3.4.3, to get the sum up value for each of the sentences of tweets. In this process we can know that which sentence are negative, neutral and positive.

3.5. Project Schedule and Milestones

The figure 3.3 show the flow of project activity from the starting step discuss project title, until the submission of final report. For the table 3.3 it have shown a gantt chart of project activity PSM1, where it is a list to follow while doing PSM1. Table 3.4 in below have shown minestrone and date PSM1, where it is a list to know when to submit progress report for supervisor during PSM. The table 3.5 it have shown a gantt chart of project activity PSM2, where it is a list to follow while doing PSM2. Table 3.6 in below have shown minestrone and date PSM2, where it is a list to know when to submit progress report for supervisor during PSM2.

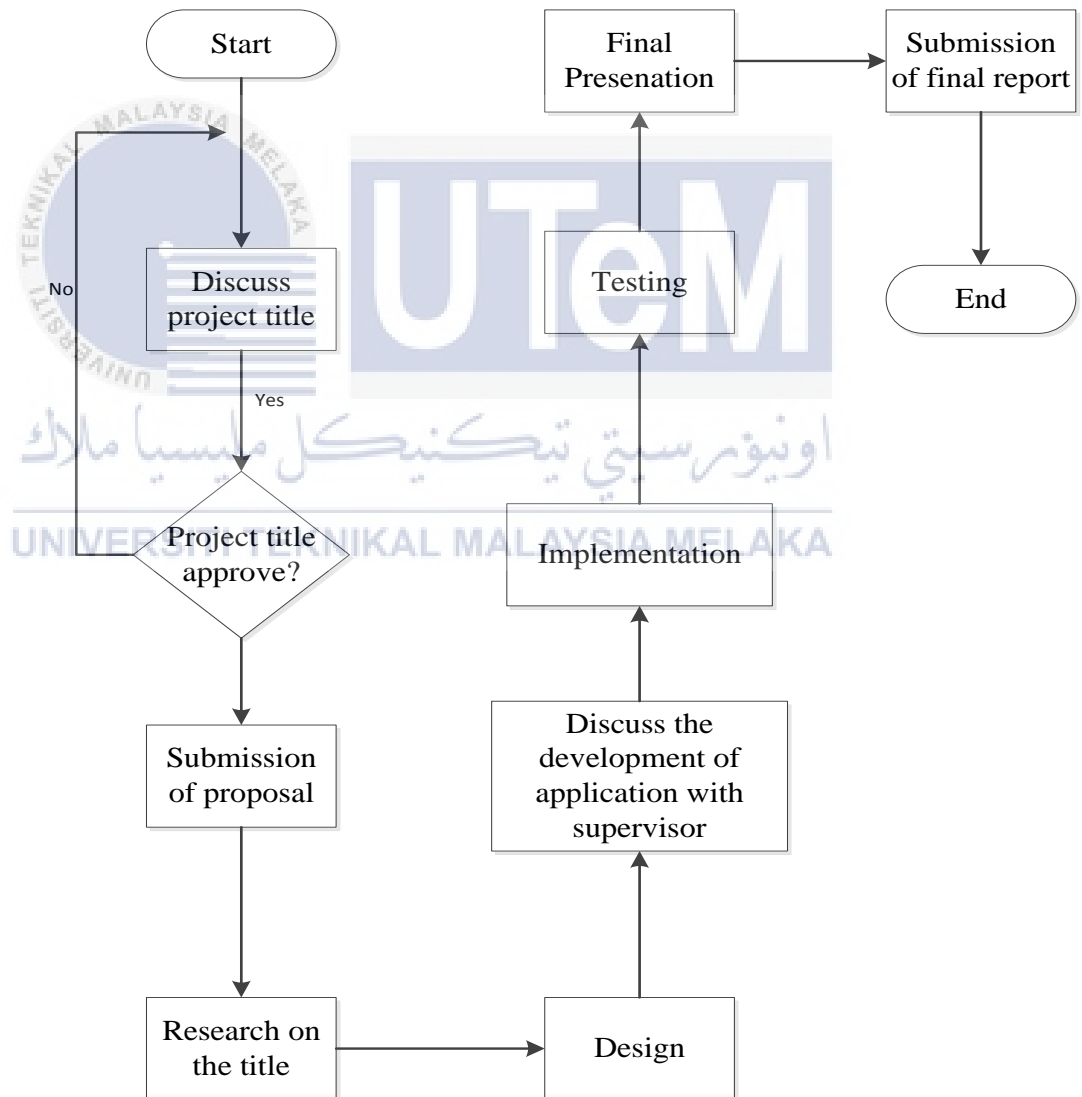


Figure 3.3 Flow chart of project activity

Table 3.3 Gantt Chart of Project Activities PSM1

No	Task Name	Week(s)														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Discussion of title and proposal submission	■														
2	Introduction		■													
3	Literature Review			■												
4	Methodology				■	■										
5	Analysis						■	■								
6	Mid-Semester Break								■	■						
7	Design									■	■					
8	Implementation											■	■			
9	Testing													■	■	
10	Presentation														■	■
11	Submission of PSM 1 Report															■

Table 3.4 Milestones and Dates PSM1

Project Activity PSM1	Duration(days)	Week(s)
Discussion of title and proposal	7	Week 1
Writing on Chapter 1 Introduction	7	Week 2
Writing on Chapter 2 Literature Review	21	Week 5

Writing on Chapter 3 Analysis	14	Week 7
Mid-Semester Break		Week 8
Design	14	Week 10
Implementation	14	Week 12
Testing	14	Week 14
Presentation	1	Week 14
Writing of PSM 1 Report	7	Week 15

Table 3.5 Gantt Chart of Project Activities PSM2

No	Task Name	Week(s)							
		1	2	3	4	5	6	7	8
1	Implementation	■	■	■	■	■	■	■	■
2	Testing				■	■	■	■	■
3	Presentation							■	■
4	Submission of Final Report								■

Table 3.6 Milestones and Dates PSM2

Project Activity PSM2	Duration(days)	Week(s)
Testing and writing Final Report	42	Week 6
Presentation	7	Week 7
Submission of Final Report	7	Week 8

3.5. Conclusion

In this chapter, we have explained about methodology and analysis phase which are important in gathering all the information and requirement to make our system better. In this chapter have shown about the flow of system. For the next chapter we will discuss about the sentiment extraction technique.



CHAPTER IV



4.1. Introduction

This chapter will explain about sentiment extraction techniques. This is important in this project because it is used to separate sentences into positive and negative sentiment. Therefore, in this chapter we will discuss about how to use Malay Word Dictionary for sentiment analysis.

4.2 Data Pre-processing

The first step of the sentiment analysis is data pre-processing mean that remove hash-tag, internet link, short form of Malay words, and all kind of emoticons that are contained in twitter. Firstly all the internet links that begin with the "http://" are removed. Next we will remove all the hash-tag "#", re-tweet "RT", colon ":", and user name start with "@". For an example RT @hyt : #Now#Holiday#Happy!, after preprocessing, it becomes *Now Holiday Happy!* Besides, we will also change all the emoticons into words for an example :(change to *sedih* and :D change to *gembira*. Word expressions like "hahahahaha", will change into "gembira" and "sob sob" will change to "nangis". All the short forms are convert into full words also, for example a word "spt", we will need to change it into "seperti". Word with too much of repeated characters and wrongly-spelled words are converted in it corresponding correct word.. Table 4.1 shows the list of processes involved in the preprocessing data

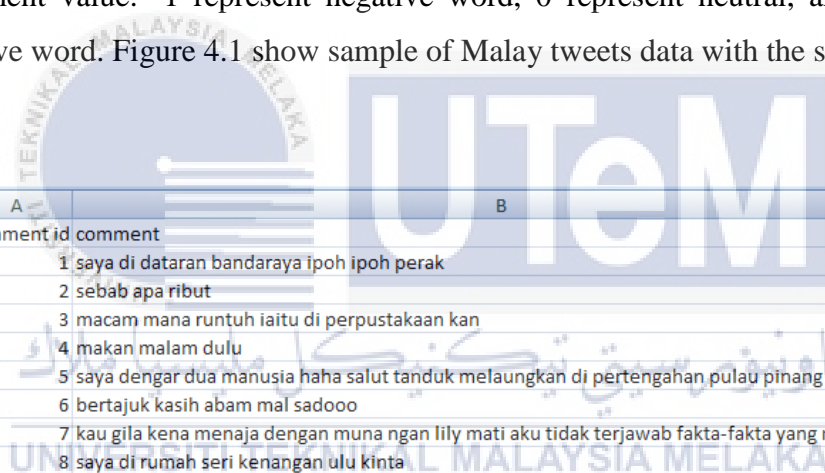
Table 4.1 Process of data pre-processing

Process	Tweet
Original Tweet data	RT @SyafiqD_Acan: Sekitar #cheliv menyambut ketibaan #mbarebellion via #allredsmalaysia #walkon #ynwa http://t.co/y7rtWV2i @noranhernandez cmne bleh runtuhhh?? dpt library kan?:(
Remove RT	@SyafiqD_Acan: Sekitar #cheliv menyambut ketibaan #mbarebellion via #allredsmalaysia #walkon #ynwa http://t.co/y7rtWV2i @noranhernandez cmne bleh runtuhhh?? dpt library kan?:(
Remove the symbol @Name	: Sekitar #cheliv menyambut ketibaan #mbarebellion via #allredsmalaysia #walkon #ynwa http://t.co/y7rtWV2i cmne bleh runtuhhh?? dpt library kan?:(
Remove hash-tag "#"	: Sekitar cheliv menyambut ketibaan mbarebellion via allredsmalaysia walkon ynwa http://t.co/y7rtWV2i cmne bleh

	runtuhhh?? dpt library kan?:(
Remove web URL	: Sekitar cheliv menyambut ketibaan mbarebellion via allredsmalaysia walkon ynwa cmne bleh runtuhhh?? dpt library kan?:(
Remove colon ":"	Sekitar cheliv menyambut ketibaan mbarebellion via allredsmalaysia walkon ynwa cmne bleh runtuhhh?? dpt library kan?:(
Change repeated words to their normal words	Sekitar cheliv menyambut ketibaan mbarebellion via allredsmalaysia walkon ynwa cmne bleh runtuh?? dpt library kan?:(
Change symbol to word	Sekitar cheliv menyambut ketibaan mbarebellion via allredsmalaysia walkon ynwa cmne bleh runtuh?? dpt library kan? sedih
Convert short form to full word	Sekitar cheliv menyambut ketibaan mbarebellion via allredsmalaysia walkon ynwa macam mana boleh runtuh?? dapat Perpustakaan?sedih
Convert English word to Malay word	Sekitar cheliv menyambut ketibaan mba pemberontakan melalui semua merah malaysia terus berjalan ynwa macam mana boleh runtuh?? dapat Perpustakaan?sedih
Remove the question mark"?"	Sekitar cheliv menyambut ketibaan mba pemberontakan melalui semua merah malaysia terus berjalan ynwa macam mana boleh runtuh dapat Perpustakaan sedih
Final clean data	sekitar cheliv menyambut ketibaan mba pemberontakan melalui semua merah malaysia terus berjalan ynwa macam mana boleh runtuh dapat perpustakaan sedih

4.3. Dictionary-based Approach

After data pre-processing, word tokenization will be performed which means that all words in the sentences will be split out one by one. When word tokenization is done, there will be a word filtering done manually where there will be adjective, adverb, noun and verb words left. After the process of extracting, all the words will be saved as a CSV file. The CSV file created named as Malay Word Dictionary will be used for word comparing when doing sentiment analysis. The next step of the process are, created a MalaySentiWordNet by tagging sentiment value for each of the word in the Malay Word Dictionary. MalaySentiWordNet are a lexical data that are created manually for opinion mining, because there still does not have a complete version of MalaySentiWordNet appear in internet or global. MalaySentiWordNet will be given each word in Malay Word Dictionary a sentiment value, where there are two type of sentiment value: -1 represent negative word, 0 represent neutral, and 1 represent positive word. Figure 4.1 show sample of Malay tweets data with the sentiment tag.



	A	B	C
1	comment id	comment	
2	1	saya di dataran bandaraya ipoh ipoh perak	neu
3	2	sebab apa ribut	neg
4	3	macam mana runtuh iaitu di perpustakaan kan	neg
5	4	makan malam dulu	neu
6	5	saya dengar dua manusia haha salut tanduk melaungkan di pertengahan pulau pinang untuk perka	pos
7	6	bertajuk kasih abam mal sadooo	pos
8	7	kau gila kena menaja dengan muna ngan lily mati aku tidak terjawab fakta-fakta yang mati bagi na	neg
9	8	saya di rumah seri kenangan ulu kinta	neu
10	9	amek cik binik lagi kerja	neu
11	10	mati sangat keraskah kena buat kajian ini	neg
12	11	majlis daerah hulu langat selangor liverbird tirtf apa lagi yang aktif sekarang	neu
13	12	apa itu majlis daerah	neu
14	13	duit hadiah yang diambil dari yuran kemasukkan tak boleh dikeluarkan yuran dikenakan hanya un	pos
15	14	aku ada dengar dua ipoh ini budak-budak tirtf agak puncak banyak pengikut dua sini namun nak me	pos
16	15	liverbird tirtf apa lagi yang aktif sekarang	neu
17	16	aku rasa aku nak perkhidmatan dibawah tawaran hangat itu masih menjadi tersebut tanya darul b	pos
18	17	tapi tapi	neu
19	18	aktif ciri itu kena iaitu persetujuan cik binik dulu kalau boleh penyenggara aku akan ciri untuk geg	pos
20	19	dan satu perkara yang aku masih pertimbangkan qadar untuk ciri aktif dalam lfc penyokong kelab l	pos

Figure 4.1 Sample of Malay tweets data with sentiment tag.

4.4. Tagging Words

The word can be tagged after going through data pre-processing. The words that are tagged are adjective and verb, done manually based on own understanding of Malay words. The double checking will be made for ensuring the words are tagged correctly. Therefore, the words tagged will be referred to the Dewan Bahasa dictionary (Malay Dictionary) to prove that the word tagged are correct. After that each of the word will be manually classified as positive, neutral, and negative, where positive word will be given a value of "1", neutral word will be assign as "0" and negative word will be given a value of "-1". The tagged words with their sentiment value will be saved as MalaySentiWordNet.

4.5. Extracting Sentiment from Tweet comments

The sentiment polarity will be calculated by summing up the positive, neutral and negative value found in the tweet comments by referring to differentiate the sentences into positive, neutral, and negative. The polarity of each of the words will be fully referred to the MalaySentiWordNet. Formula of extracting Sentiment are

$$\text{Score} = \frac{\text{Total of positive polarity} + \text{Total of negative polarity} + \text{Total of neutral polarity}}{\text{Total number of words detected}}$$

Example 1: "dekat iphone tidak ada aplikasi maya haiwan kesayangan" the adjective and verb words found are ('dekat', '1'), ('tidak', '-1'), ('ada', '1') there are two words categorize as 1 and only one word categorize as -1. Therefore, this sentences will be classified as positive sentences because there are more positive words than negative word.

Example 2: "kualiti itu stabil atau tidak" the adjective and verb words found are ('kualiti', '0'), ('stabil', '1'), ('tidak', '-1') there are one words categorize as 0, one words categorize as 1 and one word categorize as -1. Therefore, this sentences will be classified as neutral sentences because there are balance between positive, negative and neutral. Figure 4.2 shows the sample of MalaySentiWordNet.

Example 3: "jalan ini sangat sempit dan kecil" the adjective and verb words found are ('sempit', '1'), ('kecil', '1'), there are two word categorize as -1. Therefore, this sentences will be classified as negative sentences because there are only negative

words found. Figure 4.2 shows the sample of MalaySentiWordNet with words category tag of adjective.

	A	B	C	D	E
1	ID	WORD	TAG	VALUE	CATEGORY
2	100002	akhir	neg	-1	a
3	100007	terakhir	neg	-1	a
4	100008	kualiti	neu	0	a
5	100010	kencang	neu	0	a
6	100011	ketara	neu	0	a
7	100013	terhad	neg	-1	a
8	100016	rampung	pos	1	a
9	100017	sempit	neg	-1	a
10	100019	berakal	pos	1	a
11	100020	ketat	neg	-1	a
12	100021	rosak	neg	-1	a
13	100022	merdeka	pos	1	a
14	100023	dengki	neg	-1	a
15	100025	terjojol	neg	-1	a
16	100028	angkuh	neg	-1	a
17	100029	serius	neg	-1	a
18	100031	stabil	pos	1	a
19	100032	kacau	neg	-1	a
20	100038	berdarah	neg	-1	a

Figure 4.2 Sample of MalaySentiWordNet with words category tag of adjective.

4.6. Conclusion

In this chapter, we have explained about sentiment extraction technique which describe how does the system work to tag the word and calculate sentiment value for the word. This chapter also have shown the flow of system. In the next chapter, we will discuss about the experimental result.

CHAPTER V



5.1. Introduction

This chapter will explain about experimental result and analysis. In this chapter, the accuracy of the sentiment analysis will be examined by using different combination of words that extract form tweets. The formula of accuracy are :

$$\text{Accuracy} = \frac{\text{Total sentences with correct sentiment tag}}{\text{Total sentences}}$$

5.2 Experimental Result

The words will be differentiated as four categories : **adjectives** are labelled as **a**, **adverbs** are labelled as **r**, **nouns** are labelled as **n** and **verbs** are labelled as **v** as outlined in table 5.1. After all categories have been tagged, we will extract and compare the positive, negative and neutral labelled words. Therefore if the positive labelled words are more than negative words the tweet will be considered as positive, otherwise is considered as negative. If both the positive labelled words and negative labelled have same number of amount or in the sentences there are no words been extracted, it will be considered as neutral. Sentiment analysis will be implemented on the testing data using several lexical based techniques. The techniques are several combination of different words, part of speech.

Table 5.1 Category of words.

Number	Full name	Short form
1.	adjective	a
2.	adverb	r
3.	noun	n
4.	verb	v

The first type of combinations are **ar_addnv**, it has 3 main steps, where the first step is checking adjectives and adverbs in a tweet with its sentiment polarity, if no adjectives and adverbs are found then it will look for nouns, if there is no nouns found, it will go to look for verbs. If there is a count found in the either step with its sentiment polarity, the counting will stop and will classify the sentiment of the tweet.

Example 1: if the tweet contain 3 positive adjectives, 1 negative adverb and 2 verbs. First step need to check on the frequency of adjective and adverb words, there are 3 positive labelled adjective, and 1 negative labelled adverb found in the tweet. In this

example there are a count found in the first step, so the counting will stop in this step and does not need to count on for the verb. Therefore, the result for this example will be positive because the frequency of positive is more than negative.

Example 2: If the tweet contain 2 negative nouns and 1 verb. First step is to check on the frequency of adjective and adverb words. There are 0 count for adjective and adverb, so second step are needed, where it will check again for the noun labelled word, and there are 2 negative noun found. Since there a count found on second step, so we can ignore for the third step mean that does not need to check for the polarity of the verb. Therefore, the result for this example will be negative because the frequency of negative is highest. Figure 5.1 show the result of the combination of **ar_addnv** based on its polarity where positive with 57.6%, negative with 19.4%, neutral with 16.8%, and none with 6.2%. Example for the formula of getting the positive result as shown in the pie chart are
$$\text{Positive} = \frac{\text{Total sentences with positive tag}}{\text{Total sentences}}$$
.

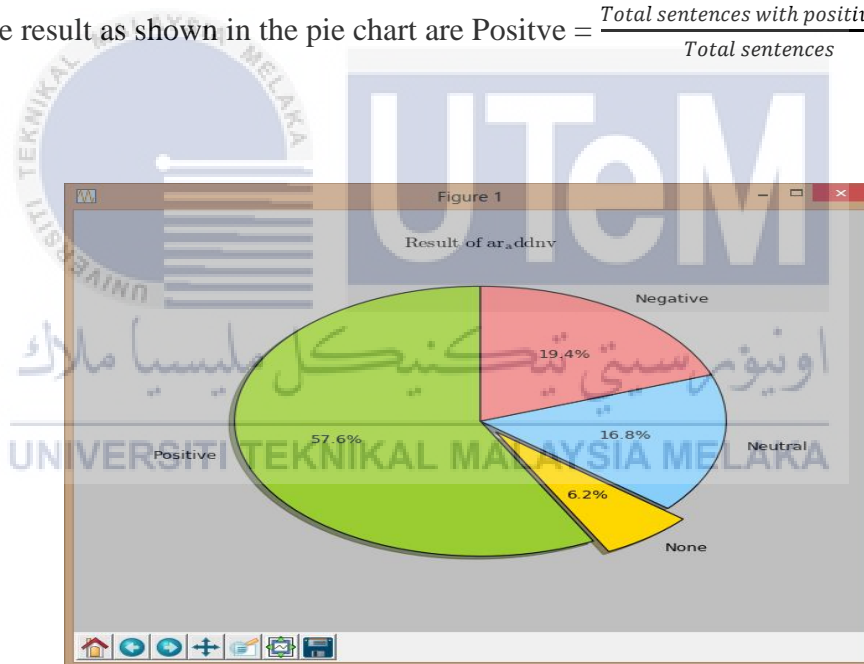


Figure 5.1 Result of ar_addnv.

Second type of combination are **ar_addvn**, it has 3 main steps, where the first step is checking adjectives and adverbs in a tweet with its sentiment polarity, if no adjectives and adverbs are found then it will for verbs, if no verbs found, it will go look for nouns. If there is a count found in the either step with its sentiment polarity, the counting will stop and will classify the sentiment of the tweet. The step for checking ar-addvn is almost same with ar-addnv, just the arrangement of second step

and third step are opposite compared to ar-addvn. Figure 5.2 shows the result of the combination of **ar_addvn** based on its polarity where positive with 57.3%, negative with 19.3%, neutral with 23.4%, and does not have none.

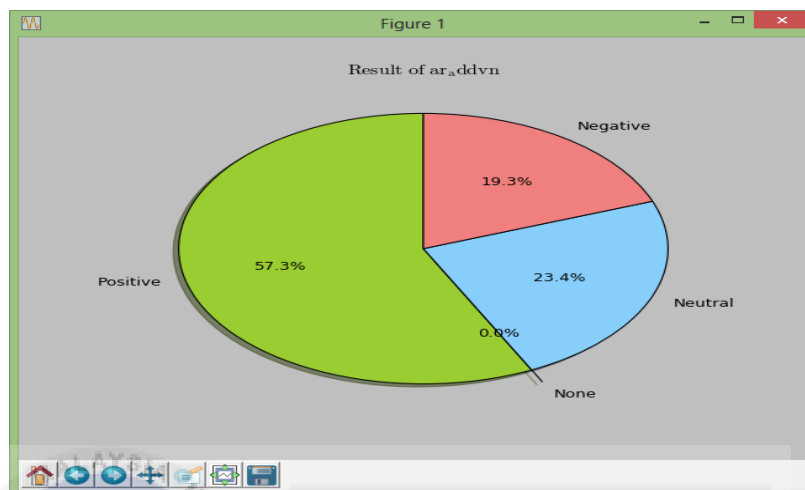


Figure 5.2 Result of ar_addvn.

Third combinations is **ar**, where in this combination it will count for the sentiment polarity of the words category of adjective, adverb together. For example, if the tweet contain 3 positive adjectives and 1 negative adverb, then the result will be positive because the frequency of positive is more than negative. Figure 5.3 shows the result of the combination of **ar** based on its polarity where positive with 52.6%, negative with 18.0%, neutral with 14.5%, and none with 14.9%.

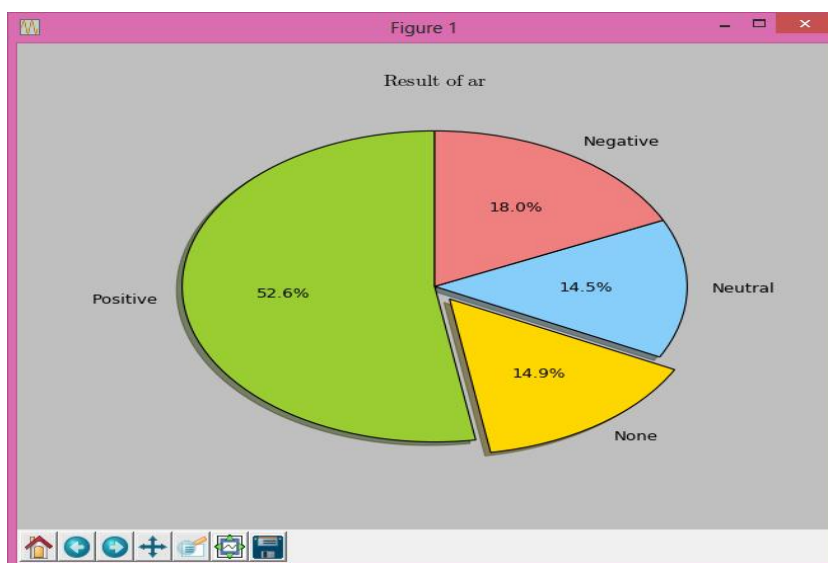


Figure 5.3 Result of ar.

Fourth combination is **arn**, where in this combination it will count for the sentiment polarity of the words category of adjective, adverb, and noun together. Figure 5.4 shows the result of the combination of **arn** based on its polarity where positive with 57.9%, negative with 17.7%, neutral with 14.9%, and none with 9.5%.

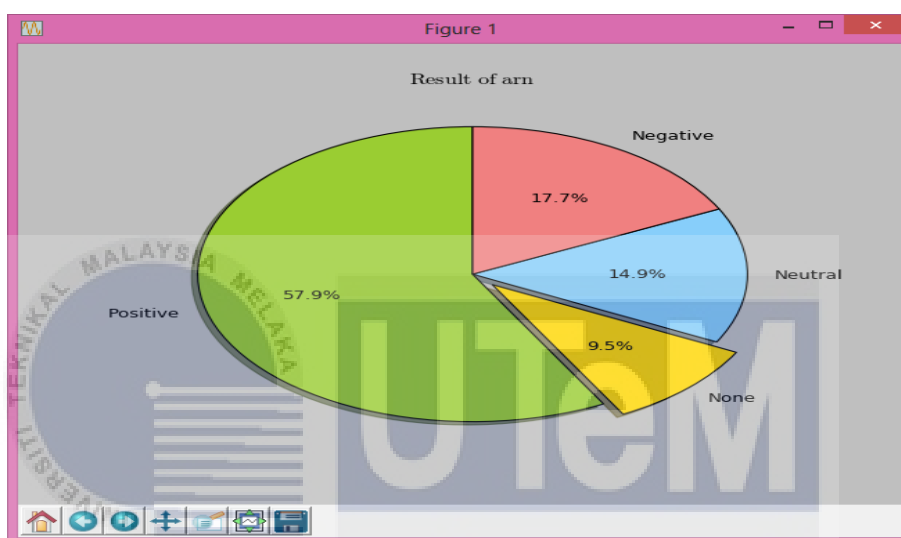


Figure 5.4 Result of arn.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Fifth combination is **arv**, where in this combination it will count for the sentiment polarity of the words category of adjective, adverb, and verb together. Figure 5.5 shows the result of the combination of **arv** based on its polarity where positive with 53.5%, negative with 18.2%, neutral with 17.6%, and none with 10.7%.

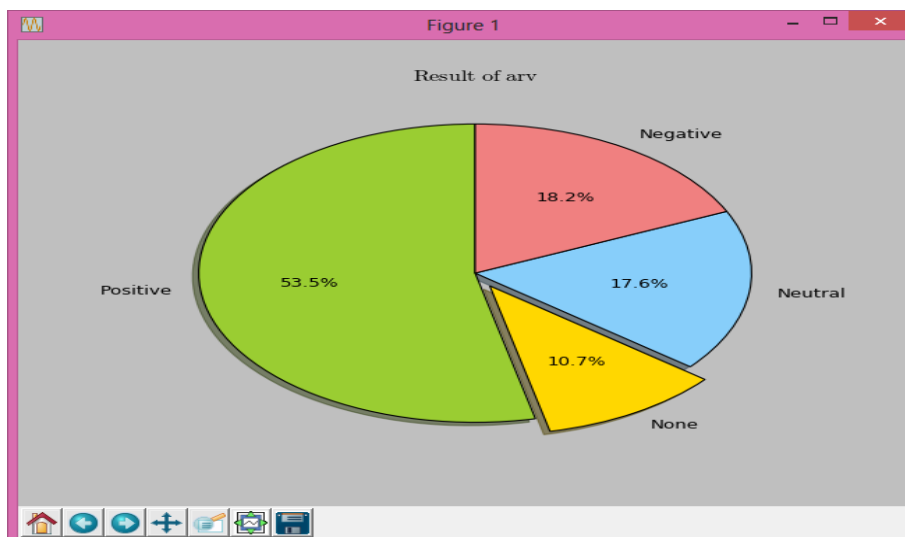


Figure 5.5 Result of arv.

Sixth combination is **nv**, where in this combination it will count for the sentiment polarity of the words category of noun and verb together. Figure 5.6 shows the result of the combination of **nv** based on its polarity where positive with 17.8%, negative with 4.2%, neutral with 14.3%, and none with 63.7%.

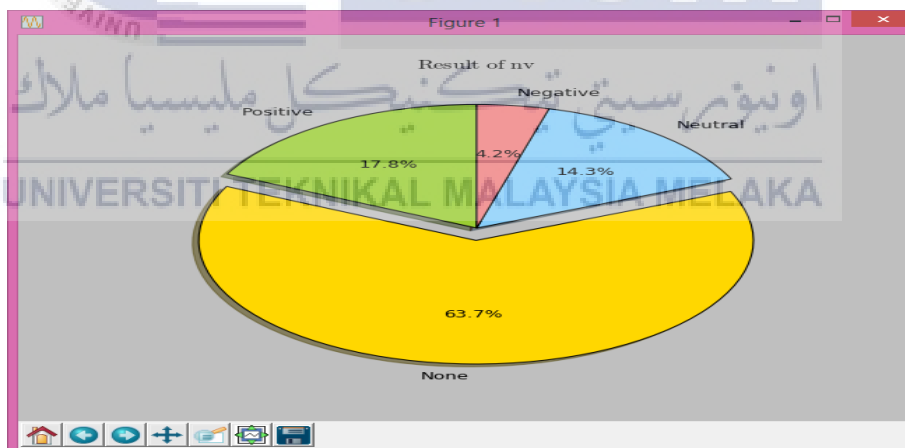


Figure 5.6 Result of nv.

Seventh combination is **arnv**, where in this combination it will count for the sentiment polarity of the words category of adjective, adverb, noun and verb together. Figure 5.7 shows the result of the combination of **arnv** based on its polarity where positive with 58% , negative with 18%, neutral with 17.8%, and none with 6.2%. Table 5.2 show the summarize of experiment.

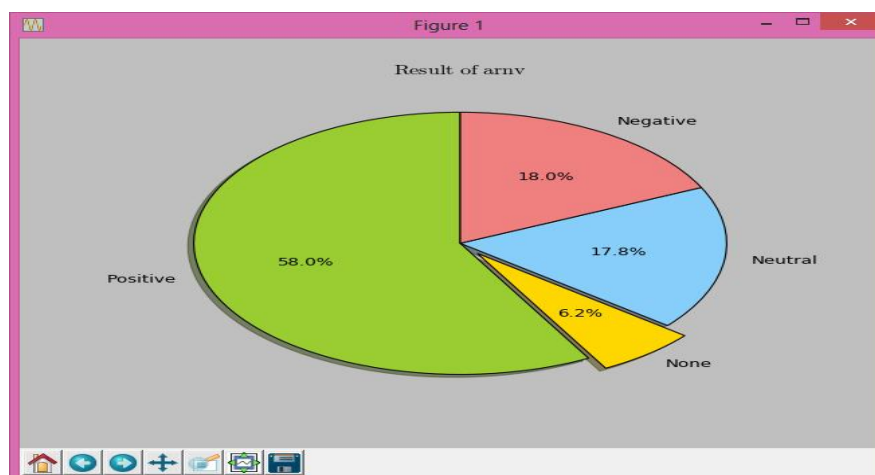


Figure 5.7 Result of arnv.

Table 5.2 Summarize of experiment

Number	Combination of words	Steps
1.	ar_addnv	<p>1) Check adjective and adverb, if have stop the process and calculate sentiment polarity.</p> <p>2) If no adjective and adverb, check noun, if the sentences contain noun stop the process and calculate sentiment polarity.</p> <p>3) If no noun check verb, then end the process and calculate sentiment polarity.</p>
2.	ar_addvn	<p>1) Check adjective and adverb, if have stop the process and calculate sentiment polarity.</p> <p>2) If no adjective and adverb, check verb, if the sentences contain verb stop the process and calculate sentiment polarity.</p> <p>3) If no verb check noun, then end the process and calculate sentiment polarity</p>

3.	ar	1) Find adjective and adverb in a sentences. 2) Calculate sentiment polarity.
4.	arn	1) Find adjective, adverb and noun in a sentences. 2) Calculate sentiment polarity.
5.	arv	1) Find adjective, adverb, and verb in a sentences. 2) Calculate sentiment polarity.
6.	nv	1) Find noun and verb in a sentences. 2) Calculate sentiment polarity.
7.	arnv	1) Find adjective, adverb, noun, and verb in a sentences. 2) Calculate sentiment polarity.

5.3. Discussion

Based on the graph below, **ar_addvn** have shown the highest accuracy compared with other techniques, where it is about 86.40% of accuracy, while for the combination of **ar_addnv** it has the second highest accuracy of 80.60%. The combination of **arnv** has the third highest accuracy compare with others which it has 80.2% of accuracy. The combination of **arn** have the second highest accuracy compared with **arnv** which it have 78.3% of accuracy, while for the combination of **ar** and **arv** have almost the same accuracy which are 72.6% and 74.9% of accuracy respectively. All lexical based techniques have high accuracy except combination of **nv** have the lowest accuracy because it does not have adjectives but only has nouns and verbs while for the most lexical based technique it contain adjective. Adjective is playing an important role in extracting sentiment analysis, because most of the adjective words have emotion. Figure 5.8 shows the accuracy of sentiment analysis based on different lexical based technique.

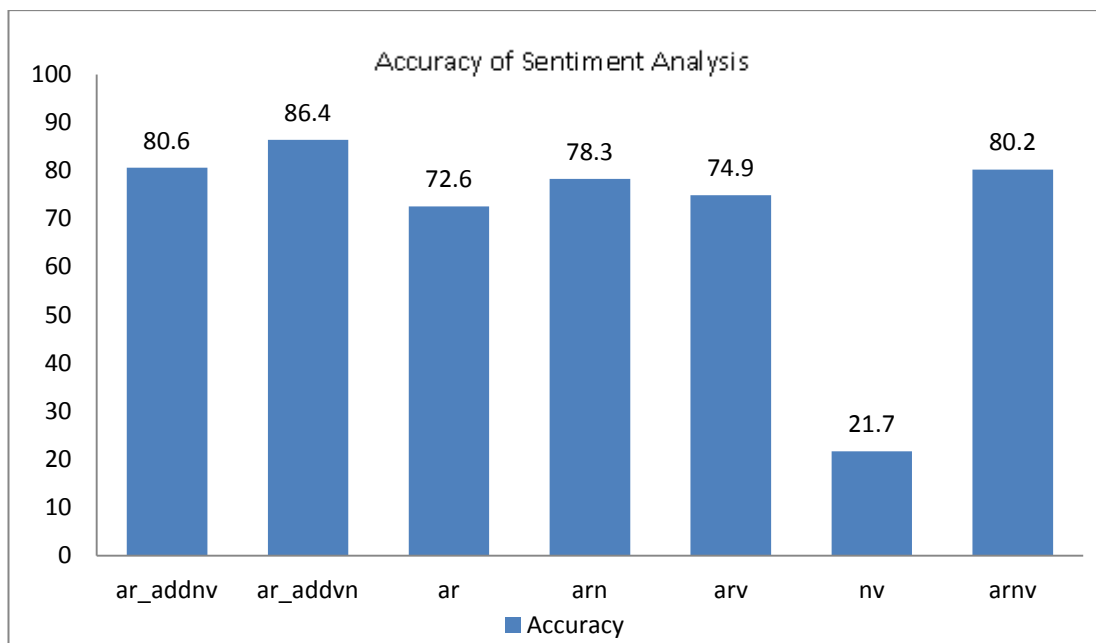


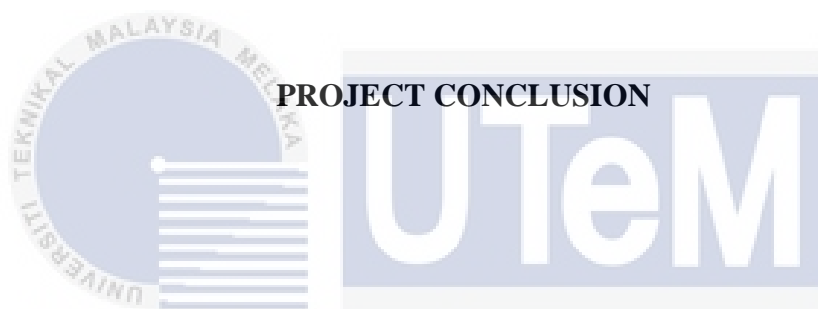
Figure 5.8 Accuracy of sentiment analysis based on different lexical based technique.



5.4. Conclusion

In this chapter, we have explained about the experiment result and analysis on different lexical based techniques. In the next chapter we will discuss about the conclusion of this project.

CHAPTER VI



6.1. Introduction

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

This chapter will explain about the project result, and will discuss for the proposition for improvement will be done for future experiment, and project contribution of Sentiment Analysis in Twitter comment.

6.2 Proposition for Improvement

In this project we have used 1000 tweet as data for sentiment analysis, but the result shown are not significant enough. Therefore for the future work, we will plan to add on some more data, to do for the training, so that we can get a more significant result. We would like to try some other combination of words based on the category for the

future work later. This is to determine which a particular combination of word category can identify more positive sentiment compared to negative sentiment.

6.3 Project Contribution

The contribution from this project are, we can learn a new technique where we do not learn before. For this project it have used Lexicon based technique, this is a kind of Natural Language Processing technique applied in Sentiment Analysis and it have occupied a nice result. Due to this project we have learn how to differentiate the comments into categories of positive, negative and neutral, this can help us a lot's in extracting feeling of human by just using words. Besides that in this project we have know that for the NLP technique of **ar_addvn** have the highest accuracy compared with other 7 techniques, which mean that this technique are more useful in determine categories of sentences. All lexical based techniques including **ar_addvn**, **ar_addnv**, **ar**, **arn**, **arv**, and **arvn** have high accuracy except combination of **nv** have the lowest accuracy because the focus of determine a sentences are adjectives that are always shown emotion.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

6.4 Conclusion

In this chapter, it had help us figure out what are the things that need to be improved for the project and master a skill of Natural Language Processing of lexicon based technique.

REFERENCES

- Poria, S., Cambria, E., Winterstein, G. and Huang, G.B., 2014. Sentic patterns: Dependency-based rules for concept-level sentiment analysis. *Knowledge-Based Systems*, 69, pp.45-63.
- Ralph, G., & Principal, I. (n.d.). Research in Text Processing: Creating Robust and Portable ... Retrieved from <http://www.aclweb.org/anthology/H90-1094.pdf>
- Pang, B., Lee, L. and Vaithyanathan, S., 2002, July. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10* (pp. 79-86). Association for Computational Linguistics.
- Khong, W.H., Soon, L.K. and Goh, H.N., 2015. A COMPARATIVE STUDY OF STATISTICAL AND NATURAL LANGUAGE PROCESSING TECHNIQUES FOR SENTIMENT ANALYSIS. *Jurnal Teknologi*, 77(18).
- Agarwal, A., Xie, B., Vovsha, I., Rambow, O. and Passonneau, R., 2011, June. Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media* (pp. 30-38). Association for Computational Linguistics.
- Xia, R., Zong, C. and Li, S., 2011. Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences*, 181(6), pp.1138-1152.
- Somprasertsri, G. and Lalitrojwong, P., 2010. Mining Feature-Opinion in Online Customer Reviews for Opinion Summarization. *J. UCS*, 16(6), pp.938-955.
- Jian, Z., Chen, X. and Wang, H.S., 2010. Sentiment classification using the theory of ANNs. *The Journal of China Universities of Posts and Telecommunications*, 17, pp.58-62.
- Go, A., Bhayani, R. and Huang, L., 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1, p.12.
- Kouloumpis, E., Wilson, T. and Moore, J.D., 2011. Twitter sentiment analysis: The good the bad and the omg!. *Icwsn*, 11, pp.538-541.
- Mullen, T. and Collier, N., 2004, July. Sentiment Analysis using Support Vector Machines with Diverse Information Sources. In *EMNLP* (Vol. 4, pp. 412-418).
- Godbole, N., Srinivasaiah, M. and Skiena, S., 2007. Large-Scale Sentiment Analysis for News and Blogs. *ICWSM*, 7(21), pp.219-222.

BIBLIOGRAPHY

Bird, S., Klein, E. and Loper, E., 2009. *Natural language processing with Python*. "O'Reilly Media, Inc."

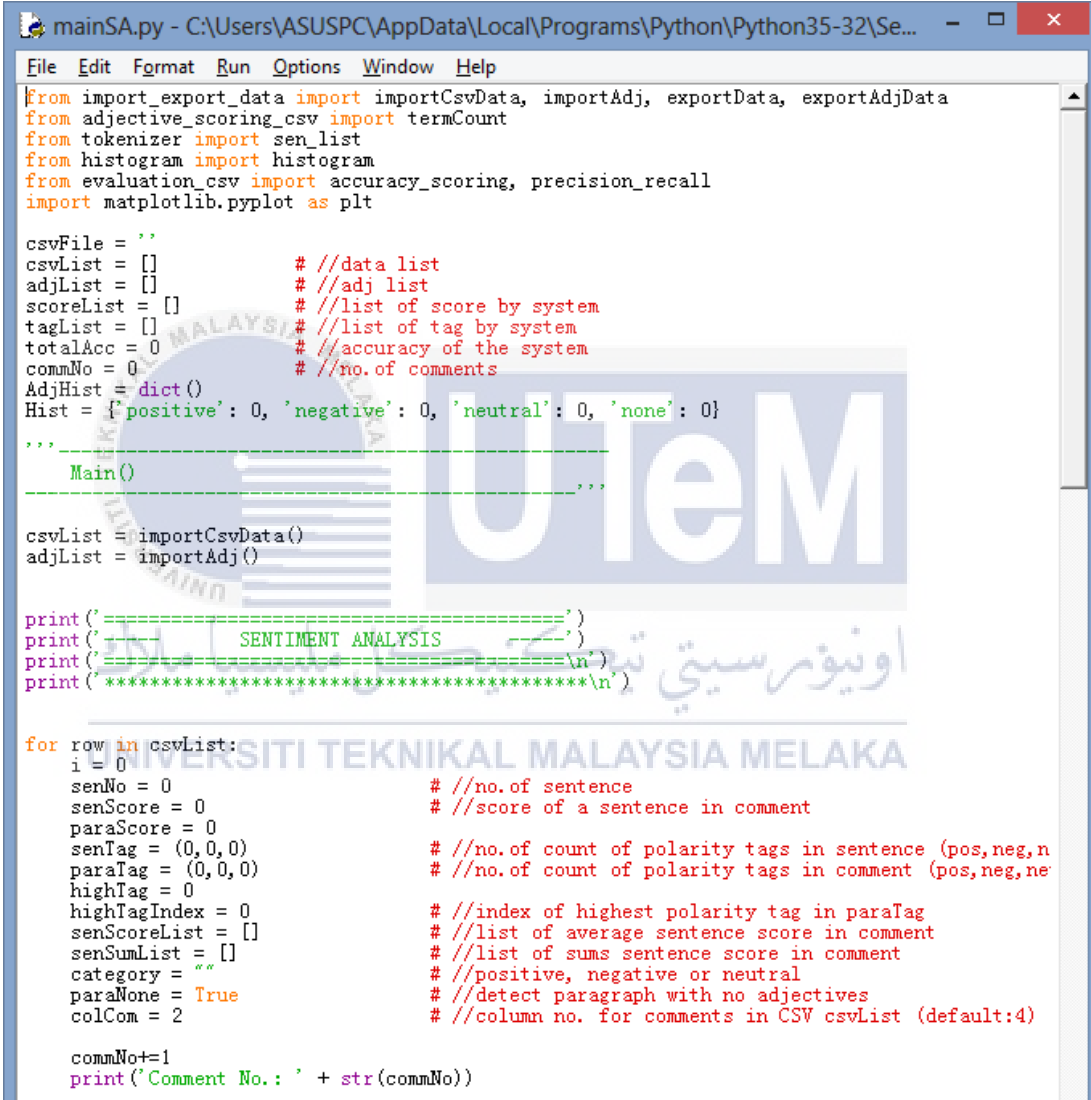
Sentiment Analysis with Python NLTK Text Classification. (n.d.). Retrieved from <http://text-processing.com/demo/sentiment/>



APPENDIX A

Coding of main part

Figure below show the main part coding on the calculation part of sentiment analysis, about how does it consider the sentences into positive, negative, neutral or balance. For an example, if there are more positive in a sentence then the sentences will be consider as positive.



```

mainSA.py - C:\Users\ASUSPC\AppData\Local\Programs\Python\Python35-32\Se... - □ ×
File Edit Format Run Options Window Help
from import_export_data import importCsvData, importAdj, exportData, exportAdjData
from adjective_scoring_csv import termCount
from tokenizer import sen_list
from histogram import histogram
from evaluation_csv import accuracy_scoring, precision_recall
import matplotlib.pyplot as plt

csvFile = ''
csvList = [] # //data list
adjList = [] # //adj list
scoreList = [] # //list of score by system
tagList = [] # //list of tag by system
totalAcc = 0 # //accuracy of the system
commNo = 0 # //no. of comments
AdjHist = dict()
Hist = {'positive': 0, 'negative': 0, 'neutral': 0, 'none': 0}

'''
Main()
'''

csvList = importCsvData()
adjList = importAdj()

print('=====')
print(' SENTIMENT ANALYSIS ')
print('=====')
print('*****\n')

for row in csvList:
    i = 0
    senNo = 0 # //no. of sentence
    senScore = 0 # //score of a sentence in comment
    paraScore = 0
    senTag = (0, 0, 0) # //no. of count of polarity tags in sentence (pos, neg, n
    paraTag = (0, 0, 0) # //no. of count of polarity tags in comment (pos, neg, ne
    highTag = 0
    highTagIndex = 0 # //index of highest polarity tag in paraTag
    senScoreList = [] # //list of average sentence score in comment
    senSumList = [] # //list of sums sentence score in comment
    category = "" # //positive, negative or neutral
    paraNone = True # //detect paragraph with no adjectives
    colCom = 2 # //column no. for comments in CSV csvList (default:4)

    commNo+=1
    print('Comment No.: ' + str(commNo))

```

```

sList = sen_list(row[colCom-1]) # //tokenize paragraph -> sentence
print(' ' + row[colCom-1] + '\n') # //preview comment
for sen in sList:
    senNo+=1

    ## //TERM-COUNTING (TC)
    dataFile = "tc"
    senTag = termCount(sen, adjList, senNo, AdjHist)

    if senTag != "None":
        paraNone = False
        paraTag = (paraTag[0]+senTag[0], paraTag[1]+senTag[1], paraTag[2]+senTag[2])

    if paraNone == True:
        category = "None"
        scoreList.append(None)
        Hist = histogram("None", Hist)
        tagList.append(None)

    else:
        if paraTag[0] == paraTag[1] and paraTag[1] == paraTag[2]:
            paraScore = 0
            category = "Balanced"
            tag = "bal"
        else:
            highTag = max(paraTag)
            highTagIndex = paraTag.index(highTag)

            if highTagIndex == 0:
                highTag = paraTag[0]
                paraScore = 1
                category = "Positive"
                tag = "pos"
            elif highTagIndex == 1:
                highTag = paraTag[1]
                paraScore = -1
                category = "Negative"
                tag = "neg"
            elif highTagIndex == 2:
                highTag = paraTag[2]
                paraScore = 0
                category = "Neutral"
                tag = "neu"

            scoreList.append(paraScore)
            Hist = histogram(paraScore, Hist)
            tagList.append(tag)

print(' POS: ' + str(paraTag[0]) + ', NEG: ' + str(paraTag[1]) + ', NEU: ' + str(paraTag[2]) + '\n')
print(' COMMENT SCORE: ' + str(paraScore) + ' (' + category + ')')
print(' *****\n')

print('----- RESULTS -----')
accuracy_scoring(scoreList, csvList)
#precision_recall(scoreList, csvList)

print('Histogram: ')
print(Hist)

# The slices will be ordered and plotted counter-clockwise.
labels = 'Positive', 'None', 'Neutral', 'Negative'
sizes = [Hist["positive"], Hist["none"], Hist["neutral"], Hist["negative"]]
colors = ['yellowgreen', 'gold', 'lightskyblue', 'lightcoral']
explode = (0, 0.1, 0, 0) # only "explode" the 2nd slice (i.e. 'Hogs')
plt.title(r'$\mathrm{Result}$ of $\mathrm{ar}$')

plt.pie(sizes, explode=explode, labels=labels, colors=colors,
        autopct='%1.1f%%', shadow=True, startangle=90)
# Set aspect ratio to be equal so that pie is drawn as a circle.

plt.show()

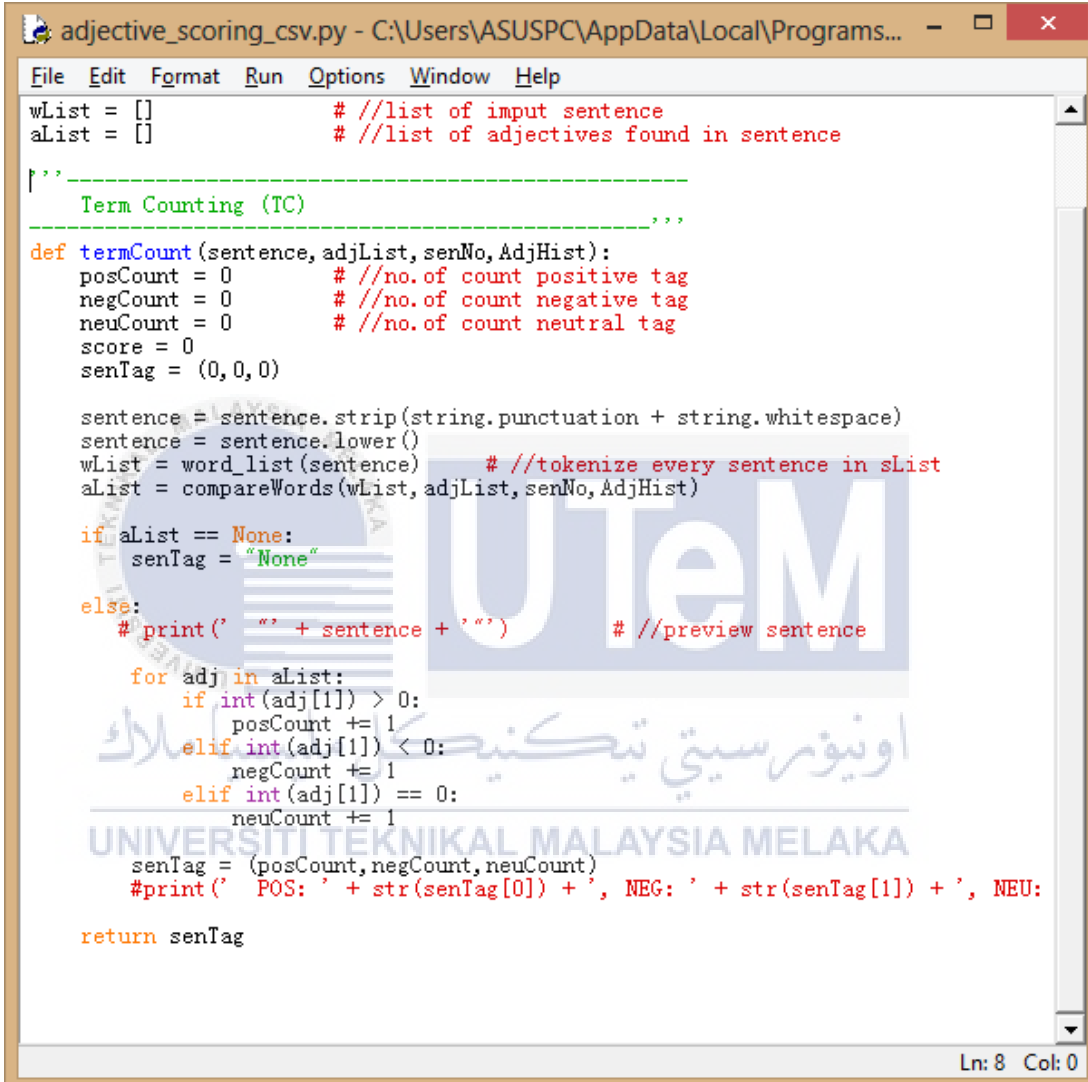
print('-----')
exportData(csvList, tagList, dataFile)
exportAdjData(AdjHist, dataFile)

```

APPENDIX B

Coding of adjective scoring

In this part it extract the adjective words and calculate the sentiment for each of the sentences.



```

adjective_scoring_csv.py - C:\Users\ASUSPC\AppData\Local\Programs...
File Edit Format Run Options Window Help
wList = []          # //list of input sentence
aList = []          # //list of adjectives found in sentence

'''
    Term Counting (TC)
'''

def termCount(sentence, adjList, senNo, AdjHist):
    posCount = 0    # //no. of count positive tag
    negCount = 0    # //no. of count negative tag
    neuCount = 0    # //no. of count neutral tag
    score = 0
    senTag = (0, 0, 0)

    sentence = sentence.strip(string.punctuation + string.whitespace)
    sentence = sentence.lower()
    wList = word_list(sentence)          # //tokenize every sentence in sList
    aList = compareWords(wList, adjList, senNo, AdjHist)

    if aList == None:
        senTag = "None"
    else:
        # print(' "' + sentence + '"')          # //preview sentence

        for adj in aList:
            if int(adj[1]) > 0:
                posCount += 1
            elif int(adj[1]) < 0:
                negCount += 1
            elif int(adj[1]) == 0:
                neuCount += 1
        senTag = (posCount, negCount, neuCount)
        #print(' POS: ' + str(senTag[0]) + ', NEG: ' + str(senTag[1]) + ', NEU:

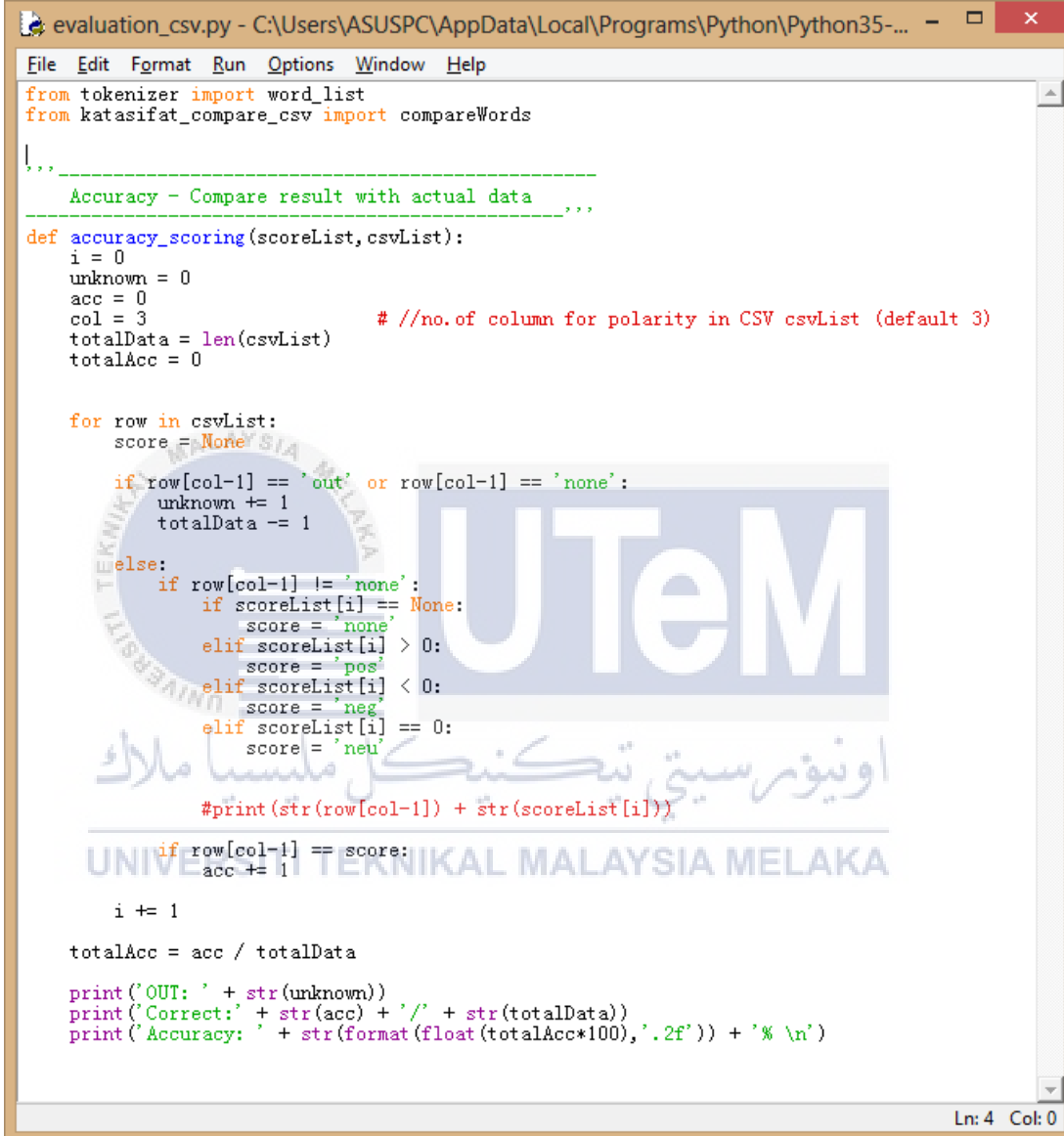
    return senTag
Ln: 8 Col: 0

```

APPENDIX C

Coding of evaluation

In this part, it will calculate the accuracy of the total result.



```

evaluation_csv.py - C:\Users\ASUSPC\AppData\Local\Programs\Python\Python35-... - □ ×
File Edit Format Run Options Window Help
from tokenizer import word_list
from katasifat_compare_csv import compareWords

'''
-----
Accuracy - Compare result with actual data
-----
'''
def accuracy_scoring(scoreList, csvList):
    i = 0
    unknown = 0
    acc = 0
    col = 3 # //no. of column for polarity in CSV csvList (default 3)
    totalData = len(csvList)
    totalAcc = 0

    for row in csvList:
        score = None
        if row[col-1] == 'out' or row[col-1] == 'none':
            unknown += 1
            totalData -= 1
        else:
            if row[col-1] != 'none':
                if scoreList[i] == None:
                    score = 'none'
                elif scoreList[i] > 0:
                    score = 'pos'
                elif scoreList[i] < 0:
                    score = 'neg'
                elif scoreList[i] == 0:
                    score = 'neu'
                #print(str(row[col-1]) + str(scoreList[i]))

            if row[col-1] == score:
                acc += 1

        i += 1

    totalAcc = acc / totalData

    print('OUT: ' + str(unknown))
    print('Correct: ' + str(acc) + '/' + str(totalData))
    print('Accuracy: ' + str(format(float(totalAcc*100), '.2f')) + '% \n')

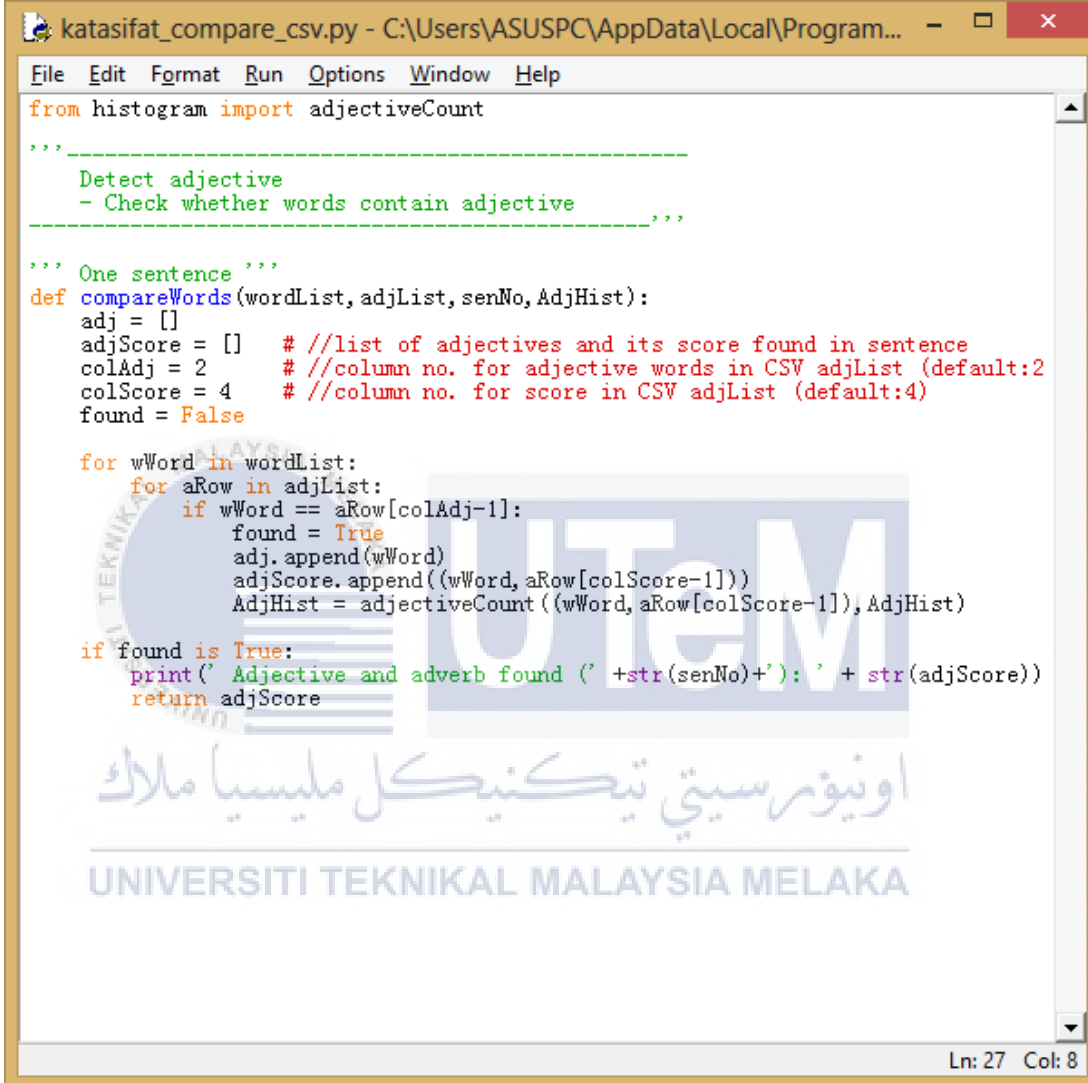
```

Ln: 4 Col: 0

APPENDIX D

Coding of katasifat compare

In this part it extract the adjective words with their sentiment tag.



```

katasifat_compare_csv.py - C:\Users\ASUSPC\AppData\Local\Program...
File Edit Format Run Options Window Help
from histogram import adjectiveCount

'''
-----
Detect adjective
- Check whether words contain adjective
-----
'''

''' One sentence '''
def compareWords(wordList, adjList, senNo, AdjHist):
    adj = []
    adjScore = [] # //list of adjectives and its score found in sentence
    colAdj = 2 # //column no. for adjective words in CSV adjList (default:2)
    colScore = 4 # //column no. for score in CSV adjList (default:4)
    found = False

    for wWord in wordList:
        for aRow in adjList:
            if wWord == aRow[colAdj-1]:
                found = True
                adj.append(wWord)
                adjScore.append((wWord, aRow[colScore-1]))
                AdjHist = adjectiveCount((wWord, aRow[colScore-1]), AdjHist)

    if found is True:
        print(' Adjective and adverb found (' +str(senNo)+'): ' + str(adjScore))
        return adjScore

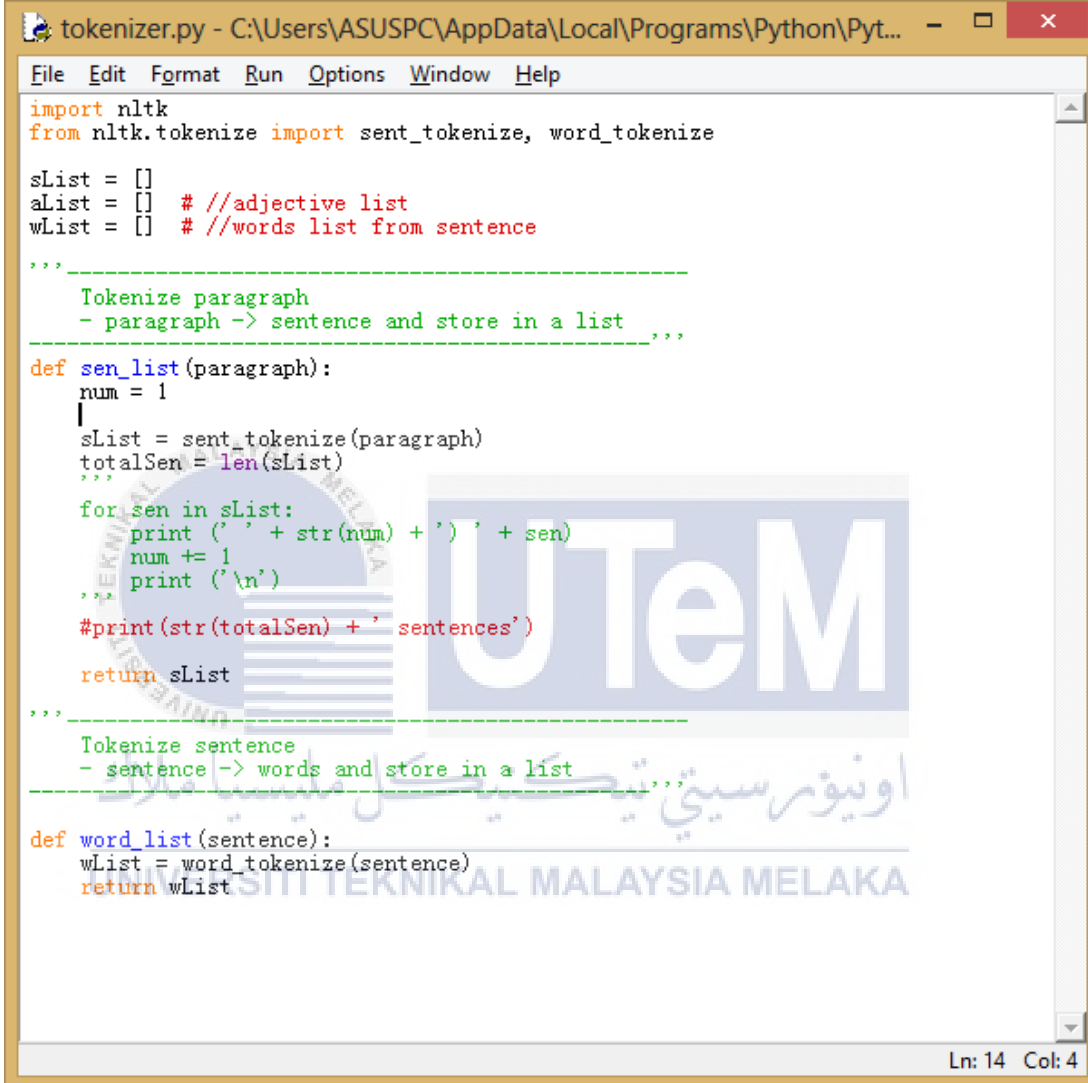
Ln: 27 Col: 8

```

APPENDIX E

Coding of tokenizer

In this part it tokenize all the words.



```

tokenizer.py - C:\Users\ASUSPC\AppData\Local\Programs\Python\Pyt...
File Edit Format Run Options Window Help
import nltk
from nltk.tokenize import sent_tokenize, word_tokenize

sList = []
aList = [] # //adjective list
wList = [] # //words list from sentence

'''
-----
Tokenize paragraph
- paragraph -> sentence and store in a list
-----'''

def sen_list(paragraph):
    num = 1
    |
    sList = sent_tokenize(paragraph)
    totalSen = len(sList)
    '''
    for sen in sList:
        print (' ' + str(num) + ' ' + sen)
        num += 1
    '''
    print ('\n')
    #print(str(totalSen) + ' sentences')
    return sList

'''
-----
Tokenize sentence
- sentence -> words and store in a list
-----'''

def word_list(sentence):
    wList = word_tokenize(sentence)
    return wList

```

Ln: 14 Col: 4

APPENDIX F

Coding of import export data

In this part cantoral of import and export of data.



```

import_export_data.py - C:\Users\ASUSPC\AppData\Local\Programs\Python\Pyth... - □ ×
File Edit Format Run Options Window Help
import os
full_path = os.path.realpath(__file__)
path = os.path.dirname(full_path)

'''
-----
  Import data from CSV file into a list
-----'''
def importCsvData():
    csvList = []          # //data list
    first = True         # //1st row of csv data

    global csvFile
    csvFile = 'try.csv'  # // shortforms converted

    openCSV = open(path + '/data/' + csvFile)

    for row in openCSV:
        if first == True: # //ignore 1st row (heading)
            first = False
        else:
            delimiter = ',' # //columns are separated by delimiter
            cell = row.strip()
            csvList.append(cell.split(delimiter))

    return csvList

'''
-----
  Import adjectives from CSV file into a list
-----'''
def importAdj():
    adjList = [] #// data list
    first = True # //1st row of csv data

    adjFile = 'adjective and adverb.csv' # //29-10-2015
    openCSV = open(path + '/data/' + adjFile)

    for d in openCSV:
        if first == True: # //ignore 1st row (heading)
            first = False
        else:
            delimiter = ',' # //columns are separated by delimiter
            cell = d.strip()
            adjList.append(cell.split(delimiter))

```

```

return adjList

'''
-----
Import adjectives score from CSV file into a list
-----'''
def importNafi():
    nafiList = []          # // data list
    first = True          # //1st row of csv data

    nafiFile = 'katanafi.csv'
    openCSV = open(path + '/data/' + nafiFile)

    for d in openCSV:
        if first == True:    # //ignore 1st row (heading)
            first = False
        else:
            delimiter = ','    # //columns are separated by delimiter
            cell = d.strip()
            nafiList.append(cell.split(delimiter))

    return nafiList

'''
-----
Export comment and tag into CSV file
-----'''
def exportData(csvList,tagList,dataFile):
    colCom = 2            # //column no. for commentid in CSV csvList (default:2)
    colTag = 3            # //column no. for comments in CSV csvList (default:8)
    i = 0

    dataheader = 'COMMENT,ORIGINAL,' + str(dataFile).upper() + '\n'
    dataFile = 'datatags-' + dataFile + '-' + csvFile
    writeCSV = open(path + '/data/' + dataFile, 'w')

    writeCSV.write(dataheader)

    for row in csvList:
        datarow = str(row[colCom-1]) + ',' + str(row[colTag-1]) + ',' + str(tagList[i]) + '
        writeCSV.write(datarow)
        i += 1

    print('Data export done!')

'''
-----
Export adjective from comments and its count
-----'''

```

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

```

def exportAdjData(AdjHist, dataFile):
    dataheader = 'WORD, TAG, COUNT \n'
    dataFile = 'adjHist-' + dataFile + '-' + csvFile
    writeCSV = open(path + '/data/' + dataFile, 'w')

    writeCSV.write(dataheader)

    for row in AdjHist:
        count = AdjHist[row]
        word = row[0]

        if row[1] == '1':
            tag = "pos"
        elif row[1] == '0':
            tag = "neu"
        elif row[1] == '-1':
            tag = "neg"
        else:
            tag = "none"

        datarow = str(word) + ',' + str(tag) + ',' + str(count) + '\n'
        writeCSV.write(datarow)

    print('Adjective and adverb export done!')
    """
    Export adjective from comments and its count
    """
    """
    View comment id and comments
    """
def viewComment():
    i = 0
    for row in csvList:
        print('Comm ID: ' + str(row[i]))
        print(row[i+3] + '\n')
    """
    View adjective list from adjFile
    """
def viewKataSifat():
    i = 0
    for row in adjList:
        print(str(row) + '\n')

```

Ln: 69 Col: 50

اوتیور سیتی تکنیکل ملیسیا ملاک

UNIVERSITI TEKNIKAL MALAYSIA MELAKA