PSM

by Muhammad Hadi Asyrafi Abdul Halim

FILE	REPORT_FYP1.DOCX (216.97K)		
TIME SUBMITTED	24-DEC-2016 01:51PM	WORD COUNT	3468
SUBMISSION ID	755942099	CHARACTER COUNT	19163

AUTOMATED REAL-TIME NEWS CLASSIFICATION SYSTEM

MUHAMMAD HADI ASYRAFI BIN ABDUL HALIM

A report submitted in partial fulfilment of
the requirements for the award of the
degree of Bachelor of Electrical – Electronics Engineering

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

December 2016

I declare that this thesis entitled " title of the thesis " is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : .....

Name : .....

Date : .....

To my beloved mother and father

ACKNOWLEDGEMENT

SDGSDFGzzbxbxbxbxgcngn

ABSTRAK

DSGSDFGDH

INTRODUCTION

1.1. Project Background

This project aims to build a running program that automatically extract news articles from websites and later classify them into categories based on classifiers trained beforehand. It will consist of three main parts which is news extraction, text preprocessing and feature generation combines together in creating a working program that can be used to classify text input. The overall coding will be written in Python as it supports related libraries for this project.

In order to extract news articles from websites, the use of APIs and RSS feeds are indeed the simplest approach available. Data accessed through this manner comes in structured form making it easier to process. However, not all websites provide these features and sometimes even if they are available they are not maintained regularly (Brody, 2012). This is where web scraping comes in. Web scraping is a technique of extracting information from websites and transforms them from unstructured form into a structured format (Ray, 2015). A Python library named 'BeautifulSoup' will be used to assist in HTML parsing. The document later will undergo HTML tags stripping which will only leave text document for further processing.

The second part of this project is text preprocessing using Natural Language Processing. "In contrast to artificial languages such as programming languages and mathematical notations, natural languages have evolved as they pass from generation to generation, and are hard to pin down with explicit rules" (Bird, Klein, & Loper, 2014). These natural language text which are in the form of unstructured data cannot be processed directly by computers (Rana, Khalid, & Akbar, 2014). Natural Language Toolkit, or NLTK in short provides rich libraries for solving the complex nature of natural language related programming in Python. Although NLP consist of many layers and stages for all kind of processing, this project will only make use of tokenization, stop-word and punctuation removal and also lemmatization.

In early days, text classification is done manually by a human indexer. It was possible those times however with the exponential increase in online materials, the task is now too demanding. This is where experts hand craft classification rules which can automatically classify documents come across but they are also deemed as difficult and time-consuming (Joachims, 2001). A machine learning approach started to emerge in place of previous methods as it more efficient and practical. There are many classification methods available such as Support Vector Machines (SVM), Naïve Bayes Classifier, Rocchio Algorithm, k-Nearest Neighbor and Decision Tree Classifier (Joachims, 2001). For this project, Support Vector Machines (SVM) a type of supervised machine learning algorithm developed by Vapnik et al. will be used. It is chosen because it is already used widely for text classification and had shown considerable results (Rana, Khalid, & Akbar, 2014) besides able to handle high dimensional input vectors from text documents (Chen & Li, 2016). However, there may be issues in determining the best kernel for SVM classification.

These three parts will be combined together into a running program which will automatically classify newly published news from several assigned websites immediately in real time. Web scraper will pass extracted news for preprocessing using NLTK library and later will be classified by SVM classifiers. The resulting classification will be cross-referenced with human based classification of the exact material for accuracy evaluation. Time taken for each classification will also be taken as one of the performance indicators of the system.

1.2. Problem statement

We have long pass the era of analog and today we can clearly see how digital the world around us is. Almost every part of our lives today is managed and handled electronically be it in communication, education, financial, military and even as far as our daily social lives. We have indeed come upon the great era of computing and this is only the beginning.

Millions of information in the form of digital data are pass around in networks spreading worldwide, reaching from one part of the world to another opposite part almost instantly. Hence, not to our surprise 90% of the world's data has only been created during these past 2 years (Dragland, 2013). This abundance of data coined as Big Data will be growing exponentially in years to come. This vast amount of data is seemingly impossible for humans to handle manually.

Businesses, governments, militaries and industries have all took part in this race. The race of extracting this large volume of data and convert them into a meaningful format. Computers are indeed exceptional in those jobs however only if they came in the form of structured data, the form of data with high degree of organizations. On the other hand, unstructured data such as text and voice are essentially the polar opposite. These unstructured data made up around 80% of the all data available (Gutierrez, 2015). These unstructured data call for Natural Language Processing.

News are an information rich source which give their consumers access to latest events and up-to-date information. In today's rapidly changing world, these knowledge gives their consumers an edge compared to others. A clear example is the stock exchange, they are extremely volatile and directly impacted by news articles published (Victor, 2011). Hence, whoever have immediate access to the news usually have the advantage compared to their peers.

1.3. Objectives

1. Use text analytics to process unstructured data from news articles
2. Create a program that can classify news as close to human based classification
3. Provides an immediate and automated classification for news from websites
4. Provides a running program that can classify news in real time
5. News classification of at least 80% accuracy

1.4. Scope of work

1. Extract news from only three websites
2. Use only bag of words model to convert unstructured data into structured data
3. Classify news into 5 categories of business, world, sport, lifestyle and others

1.5. Gantt Chart

Table 1: Gantt Chart FYP 1

No	Task	2016															
		September				October				November					December			
		1	2	3	4	1	2	3	4	1	2	3	4	5	1	2	3	4
1	Literature review																	
2	Big Data																	

3	Natural Language Processing																		
4	Machine Learning Algorithm																		
5	Web content extraction																		

Table 2: Gant Chart FYP 2

No	Task	2017																			
		February				March				April				May				Jun			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
6	Web scraper																				
7	Text preprocessing																				
8	Support Vector Machine																				
9	Application Program																				
10	Testing and tuning																				
11	Report writing																				
12	Viva preparation																				

1.6. Report Outline

This report consists of 5 chapters. *Chapter 1* covers research background as the overview of this project. Some issues that are related to this thesis were discussed in the problem statement and the objectives were stated clearly as well as the scope of this project. *Chapter 2* contains literatures reviews related to energy scavenging, piezoelectric and thermoelectric sensors. *Chapter 3* explains how this project planned to be carried out step by step. The hardware and components implementation are also described in this chapter. *Chapter 4* shows the preliminary results. Finally, conclusion will be stated in *Chapter 5*.

LITERATURE REVIEW

2.1. Introduction

Text classification has been widely researched in recent years. It is fueled by the advanced of information and communications technologies of this 20th century which makes acquiring relevant information and sorting textual data a daunting task. This topic had been approached using various methods many times before in the attempt to replace labor workforce for the task. Thankfully the advent of practical artificial intelligence algorithms had paved the path for achieving an autonomous text classification. This review will focus on recently published papers on machine learning approach for this topic.

2.2. Text Input Gathering

There are many methods available for gathering textual data for classifiers training. However, machine learning training needs a large set of data for training to get an accurate model. Some researcher relies on corpora which contains thousands of documents readily prepared for text analytics problems. Chen at el. (2016) used corpora from 20Newsgroup and RCV1-V2 while Dadgar et al. (2016) also use dataset from 20Newsgroup combined with BBC for training. The other reviewed researchers also downloaded corpus from another party such as Chinese text classification corpora of Fudan University which has 1880 classified documents (Zhou & Lili, 2010) and New China News Agency (NCNA) 2013 issues of domestic news (Cui, Meng, & Shi, 2014).

2.3. Pre-processing

Text data are unstructured data as it comes in the natural free form (Dadgar, Araghi, & Farahani, 2016). Before these unstructured data can be processed by a computer, they need to be processed beforehand. The extent of preprocessing differs between each research methodology and relies on training algorithm, feature selection and training algorithms to be used. Its objective is mainly to clean up the text from noises and useless information which may distract our classifiers. They also help in extracting only the useful information from the given text (Dadgar, Araghi, & Farahani, 2016).

Some of the most common preprocessing stages are the removal of punctuation marks, semi colons, quotes, comma, exclamation marks, date and other irrelevant characters. These characters which are defined differently across languages are known as Diacritics (Rana, Khalid, & Akbar, 2014). Some researcher also remove numbers for convenience as they also didn't contribute to text classification. Next, remaining texts are commonly broken down using tokenization into smaller segments. Tokenized texts can be represented in several levels. Unigram, bigram and trigram which are part of n-grams are the examples of sub-word level representation.

In every language, there exists stop words which are common and carries little to no meaning and are not useful for training purposes. Some example of these words in English vocabulary are 'a', 'an', 'and', 'are', 'it', and 'of' (Manning, Raghavan, & Schütze, 2008). These words are best removed to reduce training time. Fortunately, there are also prepared stop words readily available to be used provided by Journal of Machine Learning Research called SMART (Journal of Machine Learning Research, 2004). SMART contains 571 stop words commonly found in English documents (Chen & Li, 2016). In addition, there are some other preprocessing methods used such as stemming and character transformation that may be used to improve accuracy of the classifiers.

2.4. Indexing

Bag of words models are one of the simplest and most commonly used indexing methods for text classification. It treats the text document as a bag of words and convert them into term frequency representation in vector space. The downturn of this method is that it only counts the frequency of term repetition in a given text, hence syntactic and semantic are disregarded. However, this method is efficient enough in most text classification problems and gives a good accuracy.

2.5. Feature Selection

Feature selection is the process of reducing the dimensionality of feature space by selecting subsets of features for model construction (Chen & Li, 2016). There are so many types of feature selection studied before even in the scope of text classification such as Term Frequency – Inverse Document Frequency (TF-IDF), Boolean Weighting, Information Gain, Document Frequency and Mutual Information (Rana, Khalid, & Akbar, 2014). All of these studied methods shows promising results and have their own pro and cons. However, for this project feature selection are omitted. It has been reported that is indeed increases accuracy and reduces training time, but using SVM approach the differences are small.

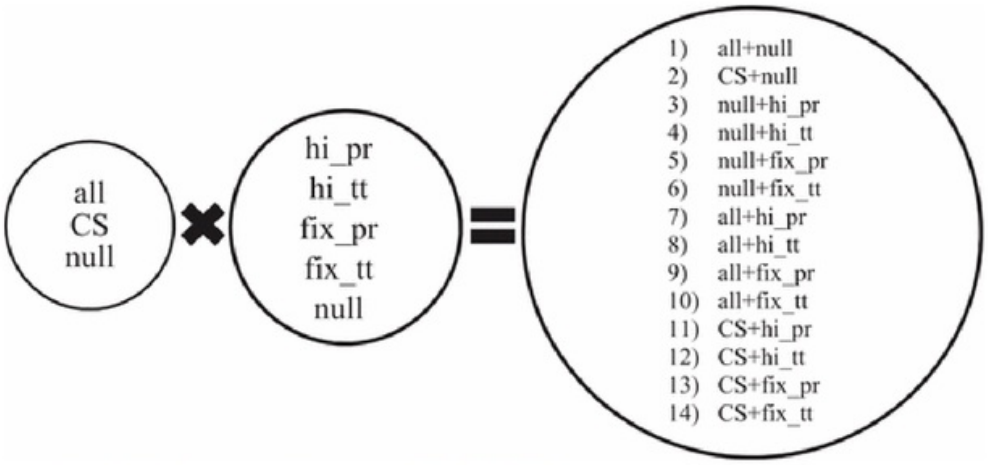

Figure 1: 14 Combination of feature sets (Chen & Li, 2016)

As mentioned in the paper by Chen et al. they conducted experiments to compare the accuracy of each combinations to find out whether implementing LDA and feature selection into SVM classification improves the result. The results are shown in figure 2 below.

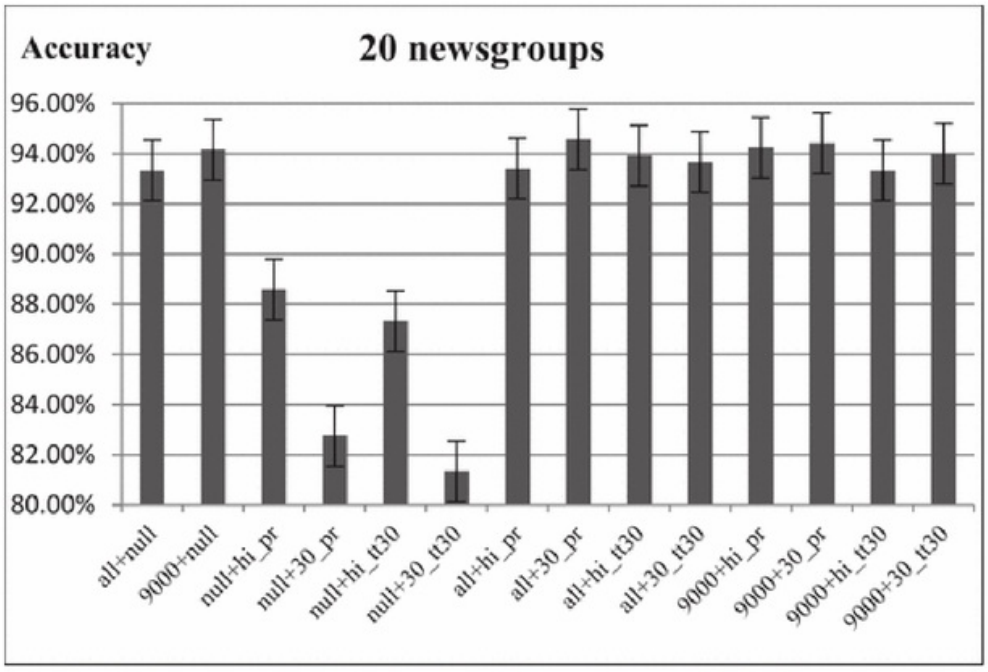

Figure 2: Accuracy of feature sets combination on 20 Newsgroups (Chen & Li, 2016)

Figure 2 above shows the differences of accuracy of classifiers trained using a combination of feature sets. '9000' refers to feature selected sets using chi-square test a feature selection method while 'all' is the experimental result of using all term without any feature selection. The term 'null' after addition symbol refers to the classifiers trained without using Latent Dirichlet Allocation (LDA). From this experimental result, we can see that even by omitting LDA and feature selection in the project we can still achieve a good accuracy by comparing 'all+null' and '9000+null'.

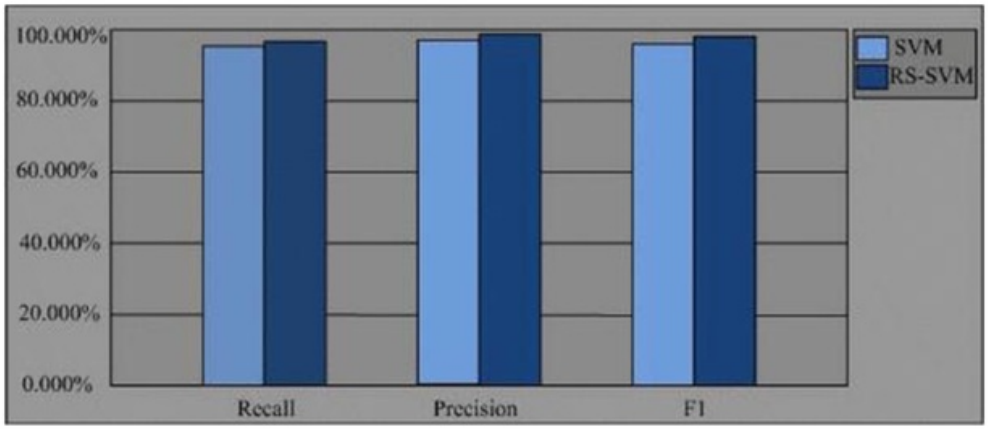

Figure 3: Recall and Precision accuracy of SVM and RS-SVM (Zhou & Lili, 2010)

Another research tries to compare the accuracy of text classification by using SVM versus SVM with feature selection using Rough Set as depicted in figure 3 also showed similar findings as mentioned before. Incorporating feature selection and LDA certainly increases the accuracy by some extent. However, this project omits both of them in trading off accuracy with simplicity.

2.6. Classification

The most important aspects for any text classification is the classification methods adopted. This plays an important role and directly affect the accuracy, precision, training and recall time. There exist many classification methods and algorithm and each has been researched before. Some examples of them are k-Nearest Neighbor, Naïve Bayes, SVM, ANN and Decision Tree (Rana, Khalid, & Akbar, 2014). From literature reviews conducted, SVM seems to be the most used algorithm in the field of text classification.

This project later will also use SVM algorithm to train classifiers as it shows promising results in previous researches. The accuracy of SVM reaches approximately 90% in all researches reviewed before. There are some researches that incorporating LDA as mentioned before. Some combines them in hierarchical methods creating 2 layers of classification (Cui, Meng, & Shi, 2014). First classifies into several most likely group using topic modelling which later will be classified into one specific category using SVM. This method of combining LDA and SVM are also used by Chen et al. in their report.

2.7. Findings

Below are the findings from 4 separate papers that have been reviewed summarized. Summary from paper authored by Rana et al. are not included as they didn't carry out any specific experiments instead focused on reviewing the most common methodologies in text classification.

Table 3: Findings from literature reviews

	(Cui, Meng,	(Chen & Li,	(Dadgar, Araghi,	(Zhou & Lili,

	& Shi, 2014)	2016)	& Farahani, 2016)	2010)
Input Corpora	New China News Agency	20Newsgroup and RCV-V2	BBC and 20Newsgroup	Chinese Text Classification Corpora of Fudan University
Preprocessing Method	Parsing, stop-word removal	Punctuation, Numbers, and SMART stop words removal. Stemming	Diachritics and stop words removal, character transforming, and tokenization	N/A
Indexing	Word sequence	Bag of Words	N/A	N/A
Feature Selection	TF-IDF	TF and or Topic Modelling	TF-IDF	Rough Set
Classification	LDA and SVM	LDA and or SVM	SVM	SVM
SVM Kernel	RBF	N/A	RBF	N/A
Accuracy of SVM classifier (approx..)	94.55%	93.00% (20 Newsgroup), 96.90% (RCV-V1)	97.84% (BBC), 94.93% (20 Newsgroup)	95.00 %

Research Methodology

3.1. Introduction

This chapter will focus on methods and approach taken to complete this project. This project focuses mainly in software components written entirely in Python programming language. We will explore extensively the idea and steps to extract news autonomously form websites, Bag of Words model for structured representation, preprocessing of text documents and also supervised machine learning algorithm called Support Vector Machine. However, this project will not delve deep into algorithm parts of these topics. Readily available open source libraries for Python will be used to help complete this project.

3.2. Project Workflow

Figure 4 shows project workflow of the Automated News Classification System from the beginning of FYP 1 until the end of FYP in general. It started with finding and determining the problem statement for this project. This will be the back bone of later phase in design and analysis.

The next step is reviewing past papers for solutions suggested by other researchers. Information gathering stage is important as it involves a lot of reading and understanding project related materials ranging from web scraping, natural language processing, artificial intelligence and machine learning algorithms that will help throughout this project.

Listing objective, aim and scope will act as a good guideline and mile stones for project design and analysis. The next stage is the development of software. The software later will be test for accuracy. If the project achieve the desired result, then the project will end with interface improvement for presentation.

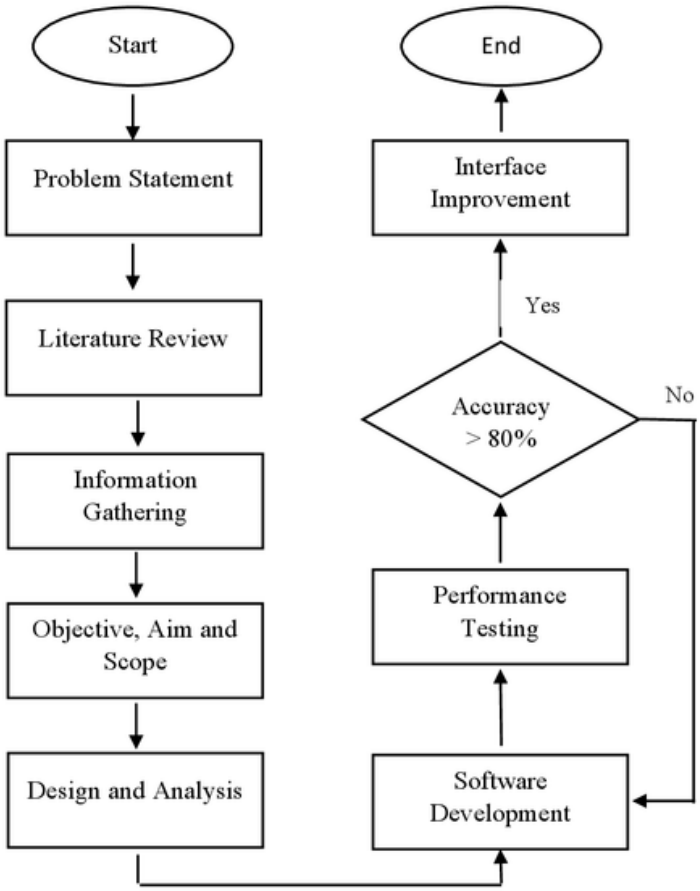

Figure 4: Project Workflow

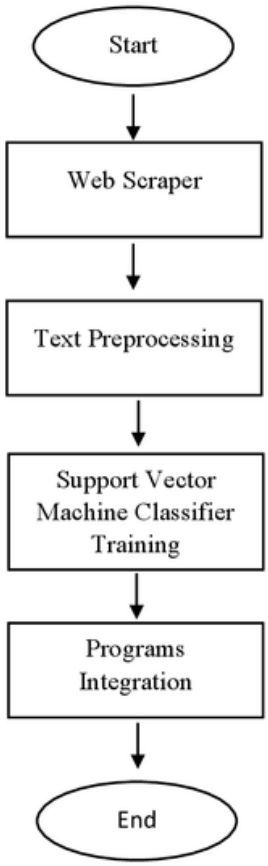

Figure 5: Software Development Flow

Figure 5 shows the development cycle of software components which are the core of this project. This project adopts the waterfall model as it is the easiest to implement and sufficient for small scale project such as this. The development part starts off with writing code for web scraper. Then, the text preprocessing part will make use of library for natural language processing. The 3rd part of this development cycle will focus entirely on training classifiers as specified in the objective. Upon completing all these 3 part, they will all combine together into a complete running program. The program development cycle will be reiterated if the accuracy of the classifiers falls below expected percentage.

3.3. Software Implementation

Software implementation is the biggest and the most important part of this project. The software development cycle I expected to take up around 65% of the total time spent on this project. As mentioned before, this project consists of 3 main parts which is web scraper, text preprocessing and classification. Each part needs to work harmoniously with each other for achieving the desired outcome.

3.3.1. Web Scraper

"Web scraping is a computer software technique of extracting information from websites" (Ray, 2015). It deploys an autonomous bot or program code that goes inside any given website refers by its URL as endpoint and extract relevant information according to the programmer specification. The information from websites are usually in HTML format. In this project, the web scraper will be programmed to automatically extract newly published news articles and feed it to the text preprocessor. The combination of 2 Python libraries which is Urllib2 which fetch the URL and BeautifulSoup to pull information will be used to perform the task.

3.3.2. Text Preprocessing

Text preprocessing is a stage to convert the textual information into a structured form so that it can be processed by computer. For processing data, this project will rely on Natural Language Toolkit an open source libraries for language processing in Python. Using this library, the preprocessing will first remove all punctuation and stop words which carries no value in term frequency analysis. Next, the remaining text will be

tokenize also using NLTK library. To reduce inflectional forms and sometimes derivationally related forms of a word to a common base form (Bird, Klein, & Loper, 2014), lemmatization is used which is also a feature inside NLTK library. Some examples are replacing 'am', 'are', 'is' into 'be' and also 'car', 'car's', 'cars' into 'car'. This will also help in reducing dimensional space that may burden the classifier training in later stage.

3.3.3. Support Vector Machines

Support is expected to be hardest part of this software development cycles. Training a lot of data and determining correct parameter settings such as kernel may directly impact accuracy. Support Vector Machines (SVM) is a powerful supervised learning classification technique based on lowest risk principle (Dadgar, Araghi, & Farahani, 2016). Its goal is to find the correct hyperplane, a plane which maximizes the margin while minimizing errors. Upon completion of classifiers training based on SVM, the location of a new input fed will be specified on the plane. Hence, deciding its class or category. An open source machine learning library for Python called scikit-learn will be used. It uses libsvm and liblinear libraires internally to handle all computations. This library also support several kernel such as RBF for classification.

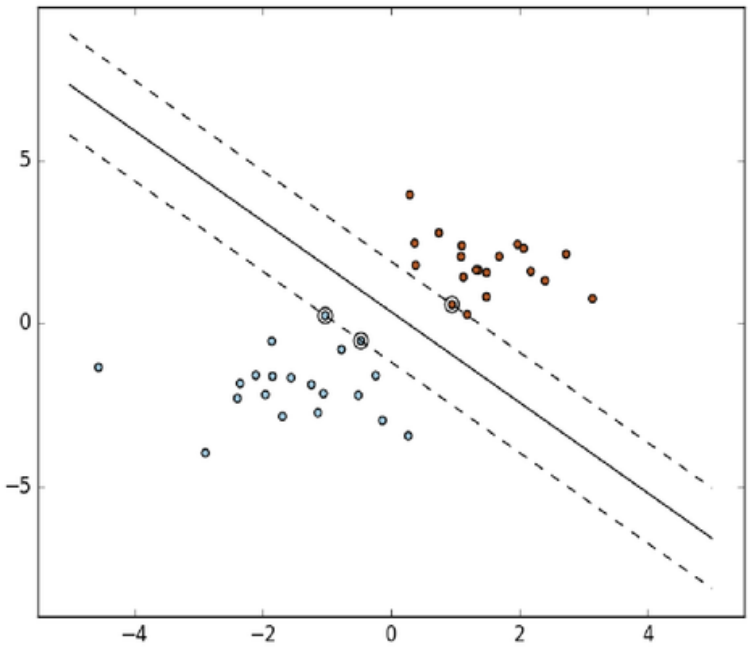

Figure 6: SVM hyperplane classification (Scikit-learn, n.d.)

CHAPTER 4

EXPECTED OUTCOME

4.1. Introduction

This chapter will discuss what is the expectation by the end of this project. What is expected to be achieved and how to determine the success of this project.

4.2. Expected Outcomes

The system should be able to automatically extract news articles directly after published on given websites and later classify the news into its respective category. This program is expected to be running in background all the time as to do classification in real time. This project can be considered as successful only if the accuracy of news classification rise above 80% of the time.

Chapter 5

Conclusion and Recommendation

PSM

ORIGINALITY REPORT

9%	7%	2%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Teknologi Malaysia Student Paper	4%
2	docs.com Internet Source	1%
3	www.macsimumnews.com Internet Source	1%
4	Submitted to University of Warwick Student Paper	<1%
5	lrd.yahooapis.com Internet Source	<1%
6	www.concentriccontent.com Internet Source	<1%
7	www.aerospaceamerica.org Internet Source	<1%
8	"Research conducted at G. Zadora and co-authors has provided new information about forensic sciences.", Biotech Week, Feb 18 2009 Issue Publication	<1%

9	www.bigdataexaminer.com Internet Source	<1%
10	portal.fke.utm.my Internet Source	<1%
11	"xiQ Announces Genpact Has Deployed Its Watson-Powered Market Intelligence Platform.", Internet Wire, May 11 2016 Issue Publication	<1%
12	f1000.com Internet Source	<1%
13	www.nlpr.ia.ac.cn Internet Source	<1%

EXCLUDE QUOTES	OFF	EXCLUDE MATCHES	OFF
EXCLUDE BIBLIOGRAPHY	ON		