

TABLE OF CONTENTS

TABLE OF CONTENT		iv
TABLE OF TABLES		vi
TABLE OF FIGURES		vii
ABSTRACT		x
CHAPTER I	INTRODUCTION	1
	1.1 Research Background	1
	1.2 Research Problem	2
	1.3 Research Scope	2
	1.4 Research Objective	3
	1.5 Research Advantage	3
CHAPTER II	LITERATURE REVIEW	4
CHAPTER III	THEORETICAL BASIS	7
	3.1 Statistical Classification	7
	3.2 Naive Bayes Classifier	7
	3.3 Term Frequency - Inverse Document Frequency (TF-IDF)	9
	3.4 Data Preprocessing	10
	3.4.1 Punctuation Removal	10
	3.4.2 Case Folding	11
	3.4.3 Stop-word Removal	11
	3.4.4 Stemming	11
	3.4.5 Tokenization	12
	3.5 K-Fold Cross Validation	12
	3.6 Performance Evaluation	12
CHAPTER IV	ANALYSIS AND SYSTEM DESIGN	15
	4.1 System Analysis	15
	4.2 System Requirement Analysis	16
	4.3 Research Steps	17
	4.3.1 Literature Study	17
	4.3.2 Data Acquisition	17
	4.3.3 System Design	18
	4.3.4 Data Preprocessing	21
	4.3.4.1 Punctuation Removal	21
	4.3.4.2 Case Folding	21
	4.3.4.3 Tokenization	22
	4.3.4.4 Stop-word Removal	22
	4.3.4.5 Stemming	23
	4.3.5 Classification Processing	24
	4.3.5.1 TF-IDF Weighting	24
	4.3.5.2 Naive Bayes Classifier	25
	4.3.6 Implementation	28

LIST OF TABLES

Table 2.1	Comparison with Related Works	5
Table 4.1	Example of punctuation removal	21
Table 4.2	Example of case folding	22
Table 4.3	Example of tokenization	22
Table 4.4	Example of stop-word removal	23
Table 5.1	Sample of dataset	34
Table 6.1	Sample of tweets extracted using the tweet extraction program	58
Table 6.2	Sample of tweets extracted with manual classification assigned	59
Table 6.3	Comparison of Raw data and Cleaned data	60
Table 6.4	Sample of tweets with each corresponding TF-IDF score	61
Table 6.5	Output of the models obtained from the training process	63
Table 6.6	Evaluation of system's performance	64

LIST OF FIGURES

Figure 3.1	Confusion Matrix (Fawcett, 2005)	13
Figure 3.2	Confusion Matrix for Multi-class Classification (Tharwat, 2018)	14
Figure 4.1	Sample of a tweet containing a hashtag #VaccinesWork	15
Figure 4.2	The Tweets classifier system architecture	19
Figure 4.3	The preprocessing architecture	20
Figure 4.4	The TF-IDF calculation architecture	24
Figure 4.5	Naive Bayes training implementation	26
Figure 4.6	Naive Bayes testing implementation	27
Figure 5.1	Implementation of Tweepy (using the hashtag #vaccinedamage)	32
Figure 5.2	Implementation of making the dataframe using Pandas	32
Figure 5.3	(A) Pandas dataframe, (B) Raw extracted data	33
Figure 5.4	Implementation of .to_csv function within Pandas	34
Figure 5.5	Implementation of the .replace() function from the String library	36
Figure 5.6	Implementation of the .lower() function	37
Figure 5.7	Implementation of the .split() function for tokenizing data	38
Figure 5.8	Result of the tokenized method	38
Figure 5.9	Implementation of stop-word removal	39
Figure 5.10	Implementation of Porter stemming	40
Figure 5.11	Python implementation of splitting the dataset using k-folding cross validation	41
Figure 5.12	(A) Process of inputting training data and re-tokenizing the dataset (B) print() of the re-tokenized dataset	42
Figure 5.13	Python implementation of the TF equation for the dataset	43
Figure 5.14	Output of the [tweet_tf] showing the result of the computeTF function	44
Figure 5.15	Python implementation of computing the count of documents containing a word	45
Figure 5.16	Python implementation of computing the IDF value of each word for each tweet in the dataset	46