

TOXIC FRIEND DETECTOR

BY

Lai Xuan Ying

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

MAY 2022

REPORT STATUS DECLARATION FORM

Title: TOXIC FRIEND DETECTOR

Academic Session: MAY 2022

I _____ LAI XUAN YING _____

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.



(Author's signature)

Verified by,



(Supervisor's signature)

Address:

No. 66, Taman Bukit Awi

09000, Kulim , Kedah

31400 Ipoh, Perak

Dr Aun Yichiet

Date: 9th September 2022

Date: 9th September 2022

Universiti Tunku Abdul Rahman			
Form Title : Sample of Submission Sheet for FYP/Dissertation/Thesis			
Form Number: FM-IAD-004	Rev No.: 0	Effective Date: 21 JUNE 2011	Page No.: 1 of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Date: 9th September 2022

SUBMISSION OF FINAL YEAR PROJECT

It is hereby certified that Lai Xuan Ying (ID No.: 18ACB05076) has completed this final year project entitle “TOXIC FRIEND DETECTOR” under the supervision of Dr Aun Yichiet (Supervisor) from the Faculty of Information and Communication Technology (FICT).

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.


Your truly,



Lai Xuan Ying

DECLARATION OF ORIGINALITY

I declare that this report entitled “**TOXIC FRIEND DETECTOR**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : Lai Xuan Ying

Date : 9th September 2022

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Dr Aun Yichiet who has given me this bright opportunity to engage this special and wonderful project on the topic Toxic Friend Detector, which also helps me in doing lots of discussion and research. It is my first step to establish a career in the Artificial Intelligence field. A million thanks to you.

Last but not least, I would also like to thank my family and friends who helped me lots of times by standing and supporting me without any condition in finalizing this project within the limited time frame.

ABSTRACT

This project is an AI design project for sentiment analysis purpose. Sentiment Analysis is the mathematically and computational study on behalf of individual attitudes, opinions and emotions expressed in written language. This research areas in natural processing, sentiment analysis and some related topic model can be measured to handle some challenging tasks like automatic personality recognition. Influence to execute he Big Five Factors Dimensions personality facets brings incremental clause in prediction the content outcomes. Several methods of natural language processing will deploy for text processing such as tokenization, POS, stemming and so on based on the input data context given. Beginning from fundamental elements of natural language processing, the outlines will keep leading o the sentiment analysis of classification. Relative to the sentiment analysis-based approach, the Lexicon-based method and machine learning: Logistic Regression, is implemented throughout the project. For some precious studies, the diversity of effects and social colorations are measured and the relationship between the personality and emotions model and the personality trait will be analysed and executed. The outcomes results will demonstrate the personality tracking that prediction and identification of the people based on the following techniques and methods which are mentioned in this project. The platform is a web-based application to retrieve the data and show the emotions or the personality of an individual as a result of prediction a toxic people.

Table of Contents

REPORT STATUS DECLARATION FORM	ii
FYP THESIS SUBMISSION FORM	iii
DECLARATION OF ORIGINALITY	iv
ACKNOWLEDGEMENTS	v
ABSTRACT.....	vi
Table of Contents	vii
LIST OF FIGURES	ix
LIST OF TABLES	xii
LIST OF SYMBOLS	xiii
LIST OF ABBREVIATIONS	xiv
CHAPTER 1: INTRODUCTION	1
1.1 Problem Statement and Motivation	1
1.2 Project Objectives	3
1.3 Project Scope	5
1.4 Impact, Significance and Contribution	6
1.5 Report Organization.....	8
CHAPTER 2: LITERATURE REVIEW	9
2.1 Review on Previous works.....	9
2.1.1 Personality and Emotional Models	9
2.1.2 Natural Language Processing	10
2.1.3 Sentiment Analysis	12
2.1.4 Feature Extraction	19
2.1.5 Strngth and weakness.....	26
2.2 Overview of research	30
2.2.1 Limitation of Previous Studies.....	30
2.3 Proposed Solution	32

CHAPTER 3: PROPOSED SYSTEM METHOD/APPROACH	35
3.1 Design Specification	35
3.1.1 Methodology	35
3.3 Timeline	41
CHAPTER 4: SYSTEM DESIGN.....	42
4.1 Project Flow Diagram	42
4.3 Speech-To-Text Procedures Flow.....	45
4.4 Natural Language Processing and Machine Learning Trained Model	53
4.5 API Flow	59
CHAPTER 5: SYSTEM IMPLEMENTATION.....	61
5.1 Hardware Setup.....	61
5.2 Software Setup	61
5.3 Setting and Configuration	66
5.3.1 Flask Configuration	66
CHAPTER 6: SYSTEM EVALUATION AND DISCUSSION	79
CHAPTER 7: CONCLUSION AND RECOMMENDATION	89
7.1 Conclusion on Project Achievements	89
7.2 Recommendation	91
BIBLIOGRAPHY	92
Appendix A: Poster.....	1
Appendix C: Final Year Project Weekly Report	1
Appendix D: Plagiarism Check Result	1
Appendix E: FYP 2 CHECKLIST	1

LIST OF FIGURES

Figure number	Title	Page
Figure 2.1	NEO-PI-RE Metric	9
Figure 2.2	Conditional Probability	15
Figure 2.3	Conditional Probability Attributes	16
Figure 2.4	Hyperplane Vector	17
Figure 2.5	Probabilistic functions	18
Figure 2.6	Bernoulli distribution	18
Figure 2.7	Regression parameters	18
Figure 2.8	Algorithm TF-IDF	20
Figure 2.9	Weight Calculation of Word Vector	20
Figure 2.10	Simple Probability Density	22
Figure 2.11	Estimation Algorithms LDA	22
Figure 2.12	Marginal Distribution	23
Figure 2.13	Probabilistic Graphical Model	23
Figure 2.14	LDA Representation Model	23
Figure 2.15	Word2Vec model architecture	24
Figure 2.16	Average log probability of Doc2Vec	25
Figure 2.17	Softmax function for probability of Doc2Vec	25
Figure 2.18	How API works	28
Figure 2.19	Agile Development Methodology	35
Figure 2.20	System Development Methodology	36
Figure 2.21	Example of Confusion Matrix	39

Figure 2.22	Gantt Chart of the proposed project development	41
Figure 4.1	Project Flow Diagram	42
Figure 4.2	Webpage Block Diagram	44
Figure 4.3	Simple Speech-To-Text Flow Diagram	45
Figure 4.4	Block Diagram for Speech-To-Text	46
Figure 4.5	MFCC Block Diagram	48
Figure 4.6	Text Classification Model Block Diagram	53
Figure 4.7	API Block Diagram	59
Figure 5.1	Flask framework application URL	66
Figure 5.2	API Configuration	67
Figure 5.3	Speech Recognition libraries 1	68
Figure 5.4	Speech Recognition libraries2	68
Figure 5.5	Code snippet of Transcription speech to text	68
Figure 5.6	Code snippet of Transcription speech to text	69
Figure 5.7	Result of Transcription speech to text	70
Figure 5.8	Export as Text Excel format	70
Figure 5.9	Transcription text in excel	71
Figure 5.10	Installation of libraries and packages	71
Figure 5.11	Data Cleaning	72
Figure 5.12	Result of NLP processing	72
Figure 5.13	Word Clouds of the toxicity of the words	73
Figure 5.14	Word Clouds of the non-toxicity of the words	73
Figure 5.15	Output of the stemmed data	74
Figure 5.16	Vectorization of TF-IDF	74
Figure 5.17	Vectorization of Word2Vec	74
Figure 5.18	Vectorization of Doc2Vec	75

Figure 5.19	Different embeddings with SMOTE	75
Figure 5.20	Normalization of DocVec	76
Figure 5.21	Algorithm classifiers models	76
Figure 5.22	Web Page of System's Web Application	77
Figure 5.22	Display Output	77
Figure 5.23	Display alert message	78
Figure 6.1	Performance measurement of every algorithm classifier with TF-IDF	79
Figure 6.2	Performance measurement of every algorithm classifier with Word2Vec	79
Figure 6.3	Performance measurement of every algorithm classifier with Doc2Vec	80
Figure 6.4	Confusion Metric of logistic regression with the Doc2Vec	80
Figure 6.5	Result of the possibly best estimator	80
Figure 6.6	Result Score of Logistic Regression with Doc2Vec	81
Figure 6.7	Result of AUC	82
Figure 6.8	Result of ROC	83
Figure 6.9	Precision-Recall Curve	83
Figure 6.10	Precision and Recall values for chosen Threshold	84
Figure 6.11	Display output at Web page	85
Figure 6.12	Display alert message at Web page	85
Figure 6.13	Nothing change after submitting the input file	86

LIST OF TABLES

Table number	Title	Page
3.1	Audio Format Settings	50
5.1	Specifications of laptop	61
5.2	Installation Python libraries	65

LIST OF SYMBOLS

β	Beta
Σ	Sigma/Sum
α	Alpha
Γ	Greek Capital Letter Gama
ξ	Greek Small Letter XI
θ	Theta
Π	Greek Capital Letter Pi

LIST OF ABBREVIATIONS

<i>AI</i>	Artificial Intelligence
<i>NEO PI-R</i>	Revised Neuroticism-Extraversion-Openness Personality Inventory
<i>OCC</i>	Ortony, Clore and Collins
<i>NLP</i>	Natural Language Processing
<i>POS</i>	Part of Speech
<i>WKWSCI</i>	Wee Kim Wee School of Communication and Information
<i>MPQA</i>	Multi-Perspective Question Answering
<i>SO-CAL</i>	Semantic Orientation Calculator
<i>NB</i>	Naïve Bayes
<i>TF-IDF</i>	Term Frequency – Inverse Document Frequency
<i>LDA</i>	Latent Dirichlet Allocation
<i>SVM</i>	Support Vector Machine
<i>RAM</i>	Random Access Memory
<i>UI</i>	User Interface
<i>LIWC</i>	Linguistic Inquiry and Word Count
<i>NLTK</i>	Natural Language Toolkit
<i>HTML</i>	Hypertext Markup Language
<i>CSS</i>	Cascading Style Sheets
<i>GUI</i>	Graphical User Interface
<i>ASR</i>	Automatic Speech Recognition
<i>HMMs</i>	Hidden Markov Models
<i>IDE</i>	Integrated Development Environment
<i>DFT</i>	Discrete Fourier transform
<i>DCT</i>	Discrete Cosine transform
<i>MFCC</i>	Mel-Frequency Cepstral Coefficients
<i>CMN</i>	Cepstral mean normalization
<i>LCP</i>	Linear predictive encoding
<i>PLP</i>	Perceptual linear prediction coefficient extraction
<i>RIFF</i>	Resource Interchange File Format
<i>WER</i>	Word Error Rate

<i>PCM</i>	Pulse-code modulation (PCM)
<i>SWER</i>	Single Word Error Rate
<i>CSR</i>	Command Success Rate

CHAPTER 1: INTRODUCTION

The title of the proposed project is Toxic Friend Detector for all young and old. This paper presents an approach for personality trait assessment in the deep learning method.

1.1 Problem Statement and Motivation

Information explosion is expeditiously rising in the social network, and it has become a central facilitator for daily communication with family members, peers, relatives, colleagues and friends. Naturally, it is easy to handle the unstructured data and information that lies behind texts which are discovered on social webs and data sources. These large quantities of content and text are dramatically convenient and useful for sentiment analysis. Sentiment analysis always tackles the computational prescription of subjectivity, opinion, behavior and sentiment of the text. Generally, a question of sentiment analysis always begins with “What other people think?” and this will lead to the issue of personality traits of an individual. People nowadays possess self-awareness when evaluating themselves in relation to others. However, people sometimes still are blinded and unaware of these kinds of toxic people. In order to prevent some tragic circumstances, they would like to filter and deal with these toxic people with some convenient ways. Perhaps there is a cautionary tool for predicting and identifying these texts' contents that those people who surround us are toxic or reliable-worthy.

The problem in sentiment analysis is classifying the polarity of the given dataset such as text conservation and sentences of other aspect levels in order to summarize the overall emotions of an individual. The performance of deep learning and various techniques will be employed for further extraction and classification. Besides, the problem is examined by building and testing several conceptual models based on the relationship between every input feature in the text contents given and the personality traits. An in-depth review of the correlations and the importance among topics sentiment analysis and natural language processing will be implemented to withstand the precision and fidelity in data analysis and data classification. Throughout this project, a cautionary and helpful tool namely Toxic Friend Detector can be proposed

CHAPTER 1: INTRODUCTION

with good accuracy to identify and analyze the toxic people during text conservation.
[1]

In the perspective of motivation research, toxic is a regularly used word for a variety of arguments and the people would like to find out some answers. In a relationship with toxic people, the victims can feel emotionally, and energy drained with a feeling of heaviness. This mischievous scenario might damage self-esteem, affect your sense of identity yet be selfless, and even cause some feeling inadequate and anxiety. In today's digital era, people are clever yet self-aware but sometimes might not predict our surrounding unseen toxic people. By reaching out here, a toxic friend detector in this project may seem helpful and essential on how to spot different kinds of toxic people. Nevertheless, this outcome can help people 'watch through' the masks people are wearing. The aim of the thesis is to propose a tool under deep learning-based algorithms modeling and natural language processing methods to deploy sentiment analysis and personality traits detection. Although their hidden identity is masked at the other sites of online devices, their original features or real thinking will be exposed by using this approach tool. Just simply say goodbye and escape their harmful behavior from toxic people. Needless to say that having a nurturing and positive relationship will boost our happiness and reduce your stress which can boost your health. A truly friend or intimate also plays a significant role in encouraging personal development and developing a sense of belonging. Through this project, we aim to have an unharmed yet safe friend who can respect each other, improve our self-confidence and enhance a healthier and happier life. In this thesis, this effective tool can be used for detecting and predicting the expressed emotions or opinions through the text conservation into some analyzed features in a way of positive, negative or neutral.

1.2 Project Objectives

The objective of this study is to propose a toxic friend detector. Although their faces and behavior are masked and unseen through online, however the conversation can be used as data input to track personality and emotions. The main objectives of this project:

i. To evaluate the natural language processing in filtering and resolving the data.

The data collected will undergo data processing to execute and filter the unnecessary and noise data. The processed data will be resolved with some technique such as tokenization and POS for data cleaning and unification. By eliminating the data content will improve the classifiers' performance and enhance the training time positively to the good result.

ii. To train and test the system by using machine learning methods based on sentiment analysis..

The data will be trained and tested by deploying hybrid methods in the combination of Lexicon- based method and Logistic regression in machine -learning methods. With the collaboration of the embeddings and normalization, the algorithms are employed to classify the content for feature extraction. The performance of these classifiers will be calculated on the basis of precision recall, f-measure and the accuracy. The outcomes will be proposed with the concept for sentiment computing.

iii. To develop a simple web application platform that can perform as a detecting personality system.

This project will build a detecting personality system in the form of a web application with a simple user interface (UI). This application will separate two alignments which the above alignment shows the column for inserting the speech and shows the speech to text content and another below alignment shows the output of toxicity detected. The content can be input as English language and the input data will be analyzed and predicted through

CHAPTER 1: INTRODUCTION

the proposed techniques and methods. An outcome of the result will be shown whether this people is toxicity or not.

1.3 Project Scope

The title of this project is Toxic Friend Detector. This project develops by using opportunistic AI to analyze and predict the personality and identify the toxic people by using speech data. The personality traits will be identified by the Big Five Dimension model in order to predict and identify an individual's emotions and personality. The scope of the project includes some phases by the following states.

Phase 1: Data Collection

In this project, only input data in the English language that is regarding social media will be taken action and consideration. The speech data will be retrieved from the surroundings friends or anyone for identification toxicity. The dialogues or the content between two individuals will predict personality traits according to the emotion model which is the Big Five Dimension model.

Phase 2: Natural language Processing

The input data will be performing a pre-processing stage to remove the stop words tokenize the text, and the stemming tokens with the NLP pipeline. This filtering process results are largely relevant to the content and increase the prediction of the system.

Phase 3: Features Engineering

Transformation words represents to the numerical version by approaching different embeddings such as term frequency-inverse document frequency (TF-IDF), Word2Vec model or Doc2Vec model. In order to encode the stemmed text, the Vector Space model (VSM) is implemented to transform the array of words into an array of numbers for the further classification tasks.

CHAPTER 1: INTRODUCTION

Phase 4: Sentiment Analysis

A hybrid model with lexicon based and machine learning based such as Naïve Bayes baseline, Linear SVC baseline and Logistic Regression baseline are conducted for sentiment analysis in the classification task. Based on the sentiment analysis in which polarity of each data context, the machine learning will detect and predict the personality traits on the given data of an individual.

Phase 5: Application Development

A web application is developed to classify personality and highlight toxic people. The main application of this application is simple yet effective dialogue detection system for an individual. The dialogues will be processed by using the toxic classification system that we proposed and will come out with prediction results.

1.4 Impact, Significance and Contribution

The goal of this project is to build an AI that can predict personality and single out toxic people by using speech data. This study has devised the Big Five model of personality dimension with merging some techniques and methods to approach this project. (Shashank Gupta, Jan 2018) Sentiment analysis is contextual mining of text which can extract and identify subjective details in the source of materials. It is the most common text classification tool to be used for detecting and analyzing an incoming message and give us information about whether it is positive, negative or neutral in the underlying sentiment. Several tools supplied by Natural language processing and machine learning along with other methods to work with large volumes of text content in order to start extracting sentiments from social media. With the use of recent advances in deep learning, some proposed algorithms and creative methods can deploy to analyze the text. An effective tool can be used creativity by the advanced artificial intelligence techniques for developing in depth research.

CHAPTER 1: INTRODUCTION

The data collection is based on users' conversation, or chit chatting where it is inadequate with some noisy and missing data. In order to improve the predictability of these text content, I will use and compare the result on predicting the outcome using different feature extraction models. Besides, several applications and techniques for machine translation, text categorization, information extraction and summarization will be executed. Classification with different kinds of machine learning algorithms is used for training and testing the data. An interpretation and comparison between the classification methods will be conducted after testing on the selected classification methods. In addition, the contribution of this study is to establish an approach that can cater the people's needs through sentiment analysis and natural language processing in social media. People can accomplish the judgments about the world surrounding them when they are living in this big environment. By developing computational linguistics and social network analysis, a best detector can be created by these contribution approaches by using the personality model, sentiment analysis, natural language processing and machine learning to have the best prediction on the personality traits.

1.5 Report Organization

The details of this research are shown in the following chapters. In Chapter 2, some related backgrounds and the literature review of the speech recognition, personality traits, sentiment analysis and the related works are reviewed. During the Chapter 3, there is a preliminary study of speech to text system with several methods and techniques. There are some several processes conducted such as data collection, data preprocessing with feature extraction, data post-processing with linguist model, decoding and result extraction and will also be presented in Chapter 3. Moreover, Chapter 4 will describe the preliminary work that had been done. For example, the setting up, data preparation, system implementation, stimulation output, data evaluation and comparison of the proposed model and the existing model. Furthermore, Chapter 5 will summarize the project with the problem tracking, motivation and proposed solutions that had be done by following chapters mentioned.

CHAPTER 2: LITERATURE REVIEW

2.1 Review on Previous works

2.1.1 Personality and Emotional Models

With the effort and help of personality and emotional models, it could widely to recognize and classify a speech of text into a predefined set of human thinking or mood. Instead of classifying these particular statements as positive, neutral and negative. These models can be more adequate to gather more information and identify the user's emotions. Needless to say that executing emotion extraction on text is a strenuous workload because our human personality and emotions are quite subjective and complex in nature. Here is the Big Five Factor model and OCC model based on personality traits are referenced in the previous work. [1]

Big Five Dimension Model

The Big Five personality traits which can served as the five- factor model (FFM) is an abstract taxonomy for personality traits in five dimensions. The five dimensions of the Big Five are OCEAN with the traits: Openness; Conscientiousness, Extroversion, Agreeableness, Neuroticism. To detect the OCEAN of every individual's personality, NEO PI-R is deployed to set a standard and former questionnaire measure of these Big Five Dimension Models. The selected NEO PI-R items which according to individual characteristics and the corresponding OCEAN-factors. These attributes can train and test for predicting the personality traits of a toxic individual. [2]



Figure 2.1: NEO-PI-RE metric

OCC Model

OCC model is developed to separate and evaluate emotions based on their underlying strategic patterns of appraisal such as the outcome one would consider appealing in any circumstance. Besides, it evaluates a classification scheme for common emotion labels that based on one's demeanour which set as positive or negative reaction to the circumstances in light of one's achievement and preferences. OCC models' classifiers the emotion into 22 emotion labels and there is an emotion generation system which formed from a set for emotion rules.

2.1.2 Natural Language Processing

Several supervisions in the form of natural language processing have been deployed by many works before preceded to the classification. In leveraging the sentiment analysis of feature extraction, text processing in natural language processing will perform to build a nature text conservation specialized in personalized personality traits. The following process can be approached to convert some plain text into processable elements with more details adjoined that can be utilized by feature extractor. Through the review on the NLP, Stop word Removal, Steeming, Tokenization and POS will be shown at the following studies. [3]

Tokenization

Tokenization is an important procedure for most NLP tasks. Tokenizer is the job of breaking the sentences and setting the output from the syntax analysis module into tokens. The token is in status that indicates either hashtag, normal word, some slang or emotion. For example, the emotions and hashtags which appear to bear much significance were restored with some exact similar word without the hash. Besides, some punctuations, elongated words and additional white spaces were corrected while the stop words were deleted. Some kinds of part of speech like nouns that are not indicative of any sentiments were discharged after POS tagging. At the last, a Porter Stemming algorithm will be used for stemming the words. For the English language part, it is minor to separate words by the spaces, however some situations such as opinion phrases that namely elements need to be considerable. The broken pieces of a

CHAPTER 2: LITERATURE REVIEW

sentence can divide into numbers, words, punctuation marks or expressions. There are several simple words like “a” “an” “the” in tokenization will be removed as these words are seemed with little help only and give very less helpful for the data. [4]

Part of speech tags (POS)

Part of Speech (POS) tags are characteristics of a word in a conversation text context and sentence based on grammatical categories of words in a language. The details of data context is important for sentiment analysis because these words may have different sentiment values based on their POS tag. For instance, the word like “bad” is a noun that encloses sentiment whereas “bad” as an adjective reflects negative sentiment. To investigate the lexical and syntactic data, the POS labeling and parsing is an effective strategy for utilization by comparing POS tags for every word. Nevertheless, there are still some successive labeling issues like word division. Several POS labels like descriptive word, thing, are very potent and helpful in light of the fact that emotion or opinions words are normally modifiers and feeling targets by the following aspects and entities from the mix thing. Moreover, parsing options syntactic data whole POS labelling gives lexical data. By comparing the POS labeling, parsing achieves weather structure data as it analyses a tree which correlates to the linguistic structure of a given sentence with the contrasting various constituents. Hence, POS labelling, parsing and some methodologies are implemented to manage a better closeness among word division.

Stop Word Removal

Stop word removal is a generally used word for instance “the”, “an”, “a”, “on” that has been programmed for search engines to evade during retrieving and searching the indexing entries for an outcome of a search query. [5] It is also being carried out in preprocessing steps throughout different kinds of natural language applications. The concept is to discharge the words that takes place usually across all the documents in perspective of corpus. Generally, the pronouns and articles are typically to be identified and classified as stop words. It is because these words are not so important and discriminative in several NLP tasks including the classification purpose and retrieving

information. [6] On the other hand, for certain NLP applications stop word removal will have very little impact but commonly the stop word list for the particular languages is well hand-curated for record of words that happens across corpuses. For examples for removing no stop words during the query processing, the text can be divided into words form a sentence and then remove those words if it occurs in the list of stop words that provided by the particular libraries such as SpaCy, NLTK and Gensim.

Stemming

Stemming is a technique to generate morphological variants of the base or a root of the word. Referring to the stemming algorithms or a stemmers, they will used to reduce the words such as “retrieving”, “retrieved”, “retrieves” and “retrieval” to diminish to the stem of “retrieve”, “Chocolates”, “choco” and so on to the root word, of the chocolate. [7] This process can affix to suffixes and prefixes and the root of words as a lemma during the information extraction and retrieval like search engines. [8] Besides, it is a significant part of the pipelining process in the Natural language processing that can discover domain vocabulaires in domain analysis. For instance, an error can minimize the words like laziness to lazi instead of lazy. However, there are some algorithms that will face some difficulties with those terms which don't look good in the mirror such as see and saw etc. There are several stemming algorithms like Porter's Stemmer algorithm which allows the suffixes by combining from simpler and smaller suffixes, Lovins Stemmer which allows to discharge the longest suffix from a word and record the word to convert the stem into valid words, and the Dawson stemmer etc.

2.1.3 Sentiment Analysis

Sentiment analysis and emotion recognition often engages attention in multiple studies such as psychology, personality traits, cognitive science, computational linguistics, text analysis and the natural language processing. Typically, sentiment analysis can evaluate to search and judge the polarity of the text towards the user's opinion. Based on the reference, it apprehends the implicit and explicit global structural details that consist of the input address in order to achieve the goal of sentiment analysis. With these talks, it can overcome the issue about the syntactic information to search the grammatical

CHAPTER 2: LITERATURE REVIEW

aspects of the statements of sentences as well as resolve a self-attention mechanism for syntactical learning. There are several approaches to address the aspect-based sentiment analysis in two conceptual tasks which are aspect sentiment classification and aspect extraction. Two types of sentiment analysis techniques specifically, the lexicon-based approach the machine-learning approach (SVM and Naïve B ayes) for predicting the text task is under which expressions or emotions of an individual. [9]

Lexicon-Based Sentiment

With the entitled states, the lexicon-based sentiment method that is set as a dictionary which accommodates thinking, emotion, mood words and their related sentiments. As expected, it is a sentiment classification that uses the orientation to measure these expressions and subjectivity in the content text. The data text is collected and processed and the sentences are parsed and tokenized. Each token is contrasted with the entities in the lexicon. While the token starts taking place in the lexicon, the correlated sentiment is positive then the overall score is increased; while the correlated sentiment is negative then the overall score is decreased. Lastly, the score is contrasted to a threshold value after the parsing is finished. For example, if the polarity is presented as positive then the score is bigger than the threshold, else the polarity will be prompt as negative. Based on the studies of Taboada et al. [10] and Thelwall et al., this sentiment lexicon method provides good yet reasonable results when applying to various kinds of domains, corpora and sentences. Besides, Taboada et al. stated, different kinds of mitigators or intensifiers modify the proper sentiment values of words separately. For example, the word “extremely” “extraordinarily” is a stronger intensifier than “rather” and sometimes the intensification may be different for a particular word which has more stronger sentiment valence than one less intense, for instance, the truly fantastic versus “truly ok”.

The SentiStrength system of Thelwall et al. committed separate scores with a scale of 1 to 5 for both positive and negative sentiments, to have assumptions that a text content can declare both positive and negative emotions. Hence, this method can be implemented by comparing different types of sentiment lexicons evaluated by comparing the sentiment valence values with simple programming in order to choose a

CHAPTER 2: LITERATURE REVIEW

suitable method. The words will have multiple parts-of-speech like occurring into adjective, verb, adverb and noun and with three main categories of positive, negative and neutral in these several sentiment lexicons such as WKWSCI, General Inquirer, MPQA Subjectivity, SO-CAL, NRC Word-sentiment Association, Hu&Liu Opinion and SentiWordNet. For example, Mohammad et al. have achieved NRC Word-Emotion Association Lexicon (EmoLex) for personality detection. EmoLex is a catalogue of English words with the correlations with several basic emotions such as terror, anger, revulsion, astonishment, delight, sorrow, belief and prediction and with two big sentiments of positive and negative. Its content was retrieved and marked manually by crowdsourcing. Furthermore, the dictionaries are built automatically or manually for lexicon-based approach while the dictionaries of words commented with the word's semantic orientation are being operated. These companion sentiment lexicons will indicate the number parenthesis and words by calculating the semantic valence score in the text content.

Machine- Learning Based Sentiment

Typically, this method is approached for learning the classification function from several number of labeled context and then deployed this classification on unlabeled content text. In order to predict a user's personality with the continues values ranging by Five-Factor model, the learning bases in classification techniques is executed. These regression models accomplish a mapping function from the feature vector to a continuous output value to analyze the polarity of the content data. Due to the machine learning implemented is supervised learning, so a training dataset is carried out necessarily. Several features such as feature selection, POS tagging, negotiations, term presence and frequency will be enforced into the list of context and opinion words. Amongst these features are also approaching the sentiment analysis as it can clarify the results of the classifiers. Through this study, the famous techniques used to indicate the learning-based analysis are Naive Bayes and Support Vector Machines. Each algorithm is different from one another however the algorithm can apply to handle the issues of sentiment categorization. (Lopamudra,D.,2016)

Naïve Bayes

Naïve Bayes (NB) is a simple statistical model that is based on Bayes Theorem which considers Naïve (Strong) independence assumption. All attributes are independent of each other which are attempted in the context of the class. This model consists of a structural model and a set of conditional probabilities. Its structural model is a directed acyclic graph in the particular nodes that serve as features or attributes while the arc will serve as attribute dependencies. Two elements of structure learning and Parameter learning resolve the classification problems by employing a classifier from a collection of labeled training data. It requires a small amount of training data to predict the significant parameters for the classification. Its classification brings practical learning algorithms and prior knowledge and discovered data can be rejoined. Hence, in Naïve Bayes method, the simple approach to analyze the probabilities of categories of the data context by using the joint probabilities of words and its personally trait categories.

For the conditional probability, it begins with the given data sentence x and class C : $P(C/x) = P(x/C) / P(x)$. Moreover, the assumption is employed for certain data point $x = \{x_1, x_2, \dots, x_j\}$, and the each of its attribute has compute the probability in a given class is independent. Thus, the probability of x can be estimated as shown $P(C/x) = P(C) \cdot \prod P(x_i/C)$. It also can be stated as:

$$P_{NB}(c|d) = \frac{P(c)(\prod_{i=1}^m P(f_i|c)^{n_i(d)})}{P(d)}$$

Figure 2.2: Conditional Probability

where the prior probability of d and c is for $P(d)$, $P(c)$ while the $n_i(d)$ is the number of times f_i which present in the context d . It is estimated that f_i is conditionally independent to every other attribute f_j so it is i is not equivalent to j ($i \neq j$) that can call the following shown values:

$$\prod_{i=1}^m P(f_i|c)^{n_i(d)}$$

Figure 2.3: Conditional Probability Attributes

When handling a crisp classification rule that assigns each instance to exactly one class, the value of the numerator for each class is calculated and choose the class which has a maximal value. This kind rule called maximum posterior rule is applied. The model that performed in figure 2 is called a simple naïve Bayes classifier.

From this case, the probability is tried to estimate that given content is under which categories of personality traits. The conditional probability that occurs can be made as a large assumption as a product of the probabilities of each word within its occurrence. This implies that there is no connection between one word to another word. The independence assumption is achieved. This study can estimate the probability of a word presenting, given the categories of emotion sentiment by seeking through a series of examples of categories personality sentiments. This is the Naïve Bayes enforces as supervised learning with the pre-classified examples for training and testing on. [14]

Support Vector Machines (SVM)

Support vector machine (SVM) is one of the popular deep machine learning systems with their outstanding and outperforming classification method at the text categorization. By contrasting to the Naïve Bayes, SVM identifies a hyperplane in its vector dimension which can divide the context as per sentiment, while the distance between the hyperplane from the nearest training-data point between this class is represented as a functional margin. Generally, the larger the margin., the lower the generalization error if the classier. This is the principle of SVM of Structural Risk Minimization. While a new unlabeled text or message is brought to the system, it draws out the feature vector in the similar manner as it applies for labeled text. The vector is applied as an input into the machine learning model. Therefore, this technique analyzes on which one side of the hyperplane which is calculated before the new datapoint prevails and a class is performed.

Due to the SVM model being two-class classification in the process whereas there are three main classes for sentiment classification, SVM tackles the issues by using one-versus all binary classification. Firstly, it is tested if the context belongs to that class or not. The process will hold on when the output is correct or else another class is tested again whether the instance belonged to one class or not. The process will continue by applying the instance to the third class when the instance is still not related to the class. The objective of this method is to search a hypothesis h for finding which error is the lowest. The solution for hyperplane vector, h can be given as:

$$\vec{h} = \sum \alpha_i C_i \vec{t}_j$$

Figure 2.4: Hyperplane Vector

The hyperplane is symbolized as h while the text is set as t , and then represents the classes into which the text has been classified as $C_j \in \{1, -1\}$ equivalent to the sentiment of the text. Whereas α_i is retrieved by solving dual optimization problems, and d_j is the support vector that devotes most to the h . Besides, the texts that have $\alpha > 0$ are the ones that commit in identifying the hyperplane. Basically, the SVMs can in charge substantial feature space with a high number of dimensions. Additionally, SVM does not presume any feature to be inconsequential. Nevertheless, the focus issue with SVM is to analyze and identify which features are more significant for classifications in order to categorize the emotions into particular classes.

The classification of SVM for Scikit learn has supported three cleanses which is SVC, LinearSVC and NuSVC which can indicate the nulit class to class classification. LinearSVC is the Linear Support Vector Classification which is quite similar to the SVC which has the kernel = “linear”. The terms different between SVC is libsvm while LinearSVC is carried out in the liblinear in order to have more flexibility in the loss functions and penalties. It is also good at scaling for huge numbers of samples. However, it does not support kernel for its parameters and attributes as determined as linear and unsufficene with some attributes like support_, support_vectors, n_support, fit_status_ and dual_coef. The purpose of LinearSVC is to fit to the data given, and categorizing and dividing data in order to return the best fit

hyperplane. Several features can be fed to the classifier after getting the hyperplane to identify what the predicted class is.

Logistic Regression

Logistic Regression (LR) is a supervised machine learning algorithm and widely implemented in various data mining and classification purposes. It is also the development of linear regression techniques whereas the outcomes are the categorical variables and response variables with one or more predictor variables. The data in the form of binary which is 0 and 1 determine that the class is from one category or another. For instance, the predictions can be held for neutral, positive and negative for the output in the situation of sentiment analysis. Thus we carry out the two functions for binary values via the logistic function and sigmoid function. [13] It can be describe as linear regression method, hence the Bernouilli distribution indicated with probabilistic functions as follows

$$f(y_i) = \pi(x_i)^{y_i} (1 - \pi(x_i))^{(1-y_i)}$$

Figure 2.5: Probabilistic functions

and $y_i = 0.1$ and for

$$\pi(x) = \frac{\exp(\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p)}{1 + \exp(\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p)}$$

Figure 2.6: Bernouilli distribution

The logit transformation (x) will occur to transform this question in order to catch up to the function $g(x)$ which is linear in its particular parameters. Hence, it is not hard to predict the regression parameters formulated as follows:

$$g(x) = \ln \left[\frac{\pi(x)}{1 - \pi(x)} \right] = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$$

Figure 2.7: Regression parameters

[14] Besides, it is known as classification algorithm so it can be categorized with some class such as binomial where it determine two types of values only and it is possible in

CHAPTER 2: LITERATURE REVIEW

the target variable “0” or “1” which can replace with some words “toxic” versus “non-toxic”, “fail” versus “pass” and “good” versus “bad” etc: Multinomial which is three or more than kins that possible in the target variables which is not arrangement which is “Bacteria A” versus “Bacteria B” versus “Bacteria C”; and the last one category that it is border categories in the related target variable, such as the movie reviews can be classified as “excellent”, “good”, “poor” and “worst” or the sequence of 0,1,2,3 and so on.

This technique can be implemented to retrieve the estimation of the unknown parameters has maximum possibilities which can evaluate the parameter of non-linear and linear models for its generality. From the mathematical calculation the least square estimation can be developed as maximum likelihood under particular normality assumptions. Furthermore it is an inductive learning algorithm that can choose an appropriate classifier for cross validation.

2.1.4 Feature Extraction

TF-IDF

For topic words extraction and generation will be introduced in TFIDF which stands for “Term Frequency – Inverse Document Frequency” (William Scoot, Feb 2019). It is applied as mathematical statistics to quantify a word in documents. These techniques will compute a weight to every word which can reflect the importance of the word to a document in a corpus or collection. In the field of information Retrieval and text mining will always deploy this method to tackle the outcomes. For example, “This girl is so beautiful”. Needless to say, it is easy for humans to understand this sentence because we all can apprehend the semantics of sentences and words. Here is the question about how the computer can understand this sentence? Hence, the vectorization of all the text can be generated with TF-IDF algorithm so that the computer can recognize the text in the form of numerical value. This method can obtain the contribution of the word’s sentiment information into the text sentiment classification. From this study, the terminology of TF-IDF can perform as *Term Frequency (TF) * Inverse Document Frequency (IDF)*, by asserted with t is term (word), d is document (set of words), N is

count of corpus and the last one *corpus* is the total document set. This method is to encounter and measure the frequency of a word in a document.

Before vectorizing the documents, the TF is individual to each document and word, hence the TF can be formulated as follows: $tf(td)$ count of t in d / number of words in d . In terms of document frequency can be measured as the essential of a document in the whole set of *corpus* whereas DF performs as the number of occurrences of term t in the document set N . It can apply with this $df(t) = \text{occurrence of } t \text{ in documents}$. This statement can retrieve the informativeness of a term and DF is the exact inverse of it. Furthermore, the inverse document frequency will be deployed to compute the informativeness of term t . When calculating with IDF, it will be very low value for to the most occurring words like stop words as the stop words such as “is” will perform in almost all f the documents and N/df will serve very low value of that word. This can bring us to the formulation of a relative weightage in $idf(t) = N/df$. Due to the equation cannot be divided by zero, the value is simplified by adding 1 to the denominator and performing to $idf = \log(N/(df + 1))$. Lastly, there are many different variations of TF-IDF, so concentrate the basic version of algorithm in sentiment analysis by following shown:

$$w(t_i, d) = \frac{tf(t_i, d) \times idf(t_i)}{\sqrt{\sum_{t_i \in d} [tf(t_i, d) \times idf(t_i)]^2}}$$

$$idf(t_i) = \log(N/n_{ti}) + 1$$

Figure 2.8: Algorithm TF-IDF

where the $w(t_i, d)$ represent the weight of the word t_i on the document d , while the $tf(t_i, d)$ represents the frequency of the word t_i in documents, N denoted the total amount of the documents and n_{ti} represent the number of documents in which the word t_i published. In this paper, where a word contains sentiment information is applied by pairing sentiment dictionaries. By the following given weight calculations method for word vectors shown as:

$$w_i = tf - idf_i \cdot e$$

$$e = \begin{cases} \alpha, & t_i \text{ is a sentiment word} \\ 1, & t_i \text{ is a non - sentiment word} \end{cases}$$

Figure 2.9: Weight Calculation of word vector

where t_i is the word, $tf-idf_i$ is the TF-IDF value of the feature word computed by equation figure 2.5 and the w_i is the weight of the words and e represents the weight based on where the word involves sentiment information. $\alpha > 1$. In this study, the TF-IDF refers to the weighted distributed word vectors with TF-IDF, which illustrates the contribution of various kinds of words to the classification task. [15]

Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is one of the probabilistic topic models that can make more inclusive assumptions on the generation of the text contrast to other techniques. [16] The evolution of LDA originally implemented to counter the problems in PLSA (Probabilistic Latent Semantic Analysis). (IR Putri and R Kusumaningrum 2017) The input of RLSA is only applied by one hyperparameter value represented as beta (β). The development LDA is achieved with the addition of hyperparameter alpha (α), and then latent variable instances which are represented as theta (θ) and phi (ϕ) are determined too. Hence, the number of parameters in LDA does not increase due to the size and volume of train corpus. Moreover, LDA method splits into two main distinct processes which is generative statistical model and inference process. LDA is a three-level hierarchical Bayesian model, in any instance of a collection is casted as a note combination over an underlying a series of topic. The study noted that LDA supervised method can execute topic words of interest for user-generated input content text and then extract them abstract when LDA is set up as an unsupervised learning method. Hence, LDA can derive the abstract words such as words of interest efficiently and effectively. It also can build up the corpus data or a set of content long text or document from latent variable which already shown. Otherwise, the LDA stated as inference process which have been characterized by the availability of corpus variable as observed data to retrieve latent data involving the word distribution over topic (ϕ) and the topic proportion for every text content or document (θ).

(David M. Blei, Andrew Y. Ng and Michael I. Jordan. 2003) The fundamental concept is that documents are represented as random mixtures over latent topics, and then characterized each topic by a distribution over words. Assumption that LDA handles as a generative process for evaluating document w in a corpus D with select N

$\sim \text{Poisson}(\xi)$ and $\theta \sim \text{Dir}(\alpha)$. The conditions of set each of the N words w_n are selecting topic $z_n \sim \text{Multinomial}(\theta)$ and a word w_n from $p(w_n | z_n, \beta)$, a multinomial probability conditioned on the topic z_n . After these few basic assumptions build in this model, several of which we discharge in subsequent sections. The Dirichlet is an effective and convenient distribution on the simplex which in the exponential family by having limited dimensional adequate statistics and conjugate to the multinomial distribution. First and foremost, the dimensionality k of the Dirichlet distribution and so the dimensionality of the topic variable z is considered stated and fixed. Also, the word probabilities are parameterized by a $k \times V$ matrix β where $\beta_{ij} = p(w^j = I | z^i = I)$, in order to estimate as a fixed quantity. Lastly, state that N is independent of all the other substance generating variables (θ and z). It can represent an ancillary and its randomness can disregard in the subsequent development. A k -dimensional Dirichlet random variable can apply the values in the $(k-1)$ -simplex (a k -vector lies in the $(k-1)$ -simplex if $\theta_i \geq 0$, $\sum_{i=1}^k \alpha_i \theta_i = 1$) and carry out the simplex probability density which is shown by the following:

$$p(\theta | \alpha) = \frac{\Gamma(\sum_{i=1}^k \alpha_i)}{\prod_{i=1}^k \Gamma(\alpha_i)} \theta_1^{\alpha_1-1} \dots \theta_k^{\alpha_k-1}$$

Figure 2.10: Simplex Probability Density

The following shows that the parameter is a k -vector with the components $\alpha_i > 0$ and states the Gamma function in $\Gamma(x)$. Referring to the latent multinomial variables in the LDA model as topics in order to manipulate text-oriented institutions however the epistemological cannot be claimed as these latent variables apart from their utility in depicting probability distributions on sets of words. These properties given will assist the progress of inference and parameter estimation algorithms for LDA. Apply the parameters, the joint distribution of a topic mixture θ , a set of N topics z , and a set of N words w is when at the following:

$$p(\theta, z, w | \alpha, \beta) = p(\theta | \alpha) \prod_{n=1}^N p(z_n | \theta) p(w_n | z_n, \beta)$$

Figure 2.11: Estimation Algorithms LDA

CHAPTER 2: LITERATURE REVIEW

where $p(z_n | \theta)$ is simply θ_i for the unique z_n^i such that $\sum_i x_{ni} = 1$. Merging over θ and counting over z , the marginal distribution of a document will be applied by the following shown:

$$p(\mathbf{w} | \alpha, \beta) = \int p(\theta | \alpha) \left(\prod_{n=1}^N \sum_{z_n} p(z_n | \theta) p(w_n | z_n, \beta) \right) d\theta.$$

Figure 2.12: Marginal Distribution

Last but not least, the product of the marginal probabilities of single documents is hold and apply the probability of a corpus as shown the following:

$$p(D | \alpha, \beta) = \prod_{d=1}^M \int p(\theta_d | \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} | \theta_d) p(w_{dn} | z_{dn}, \beta) \right) d\theta_d.$$

Figure 2.13: Probabilistic graphical model

Probabilistic graphical model shown has been represented as the LDA model.

With the figure 2.11 states it has been made clear that three levels are applied to the LDA representation. Under this model, the long content text or documents can be correlated with multiple topics. For more understanding of this model, the graphical model representation of LDA is shown. The boxes are “plates” performing replication while the outer place performing as documents (content text), besides the inner plate denotes the duplicated choice of topics and words within a document.

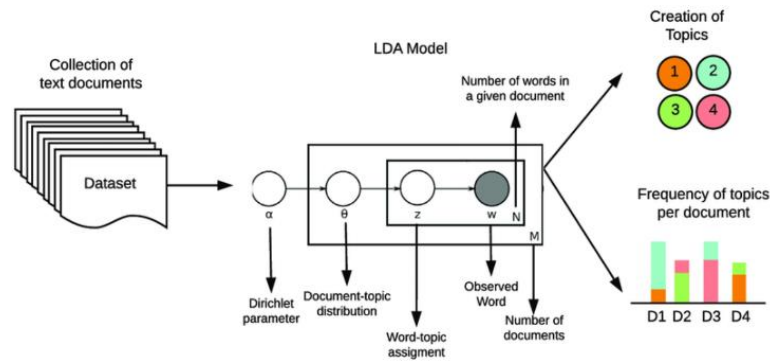


Figure 2.14: LDA representation model

Word2Vec Model

In order to do text processing, analysis and classification, [17] Word2Vec has contributed to improve the quality of the word representations in vectorization. It is also one of the propositions of word embedding that quantifies the words to overcome the semantic relations for the computer to “understand” these natural languages. The core concept of this model is to summarize the sentences composed by words that the computer anticipates are unrelated to each other into the particular higher dimensional matrix, and take over the semantic relations between words with the mathematical relations in the matrix. Thus computers can accomplish the effect of implementing the words in the same contexts which have the same vectors. It has two training models which are CBOW: Continuous Bag Words-of-Words) and Skip grams(Continuous Skip-gram Model).

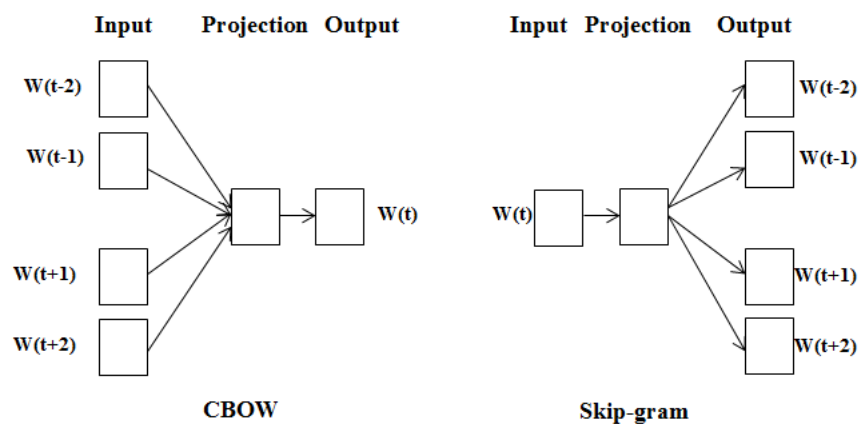


Figure 2.15: Word2Vec (CBOW and Skip-gram) model architecture

[18] For the CBOW model, the context predicts W_t for the word vectorization and the $2n$ words predicts word vector other than W in the set of S which is a superposition in the neighboring of $2n$ words vectors. The words along with the same context will have the same word vectors after a few multiple iterations whereas the principle has shown in the figure of the overview of Word2Vec on the left side ($n=2$).

For the Skim-gram model, it is quite much alike to the CBOW model. It is implemented for using the current word to predict the context. For instance a word W_t will be predicted to a context vetor and superimpose this word vector on the adjacent words. The words with the same context will have the same word vectors after multiple

interactions whereas the principle has shown in the figure of the overview of Word2Vec on the right side (n =2).

Doc2Vec Model

[19] Based on Thomas Mikolov, Doc2Vec model is an advancement and improvement according to the Word2Vec model. It is also using for embedding words and represented as vectors from the documents to predict the target words. [20]. The length of the document is not counted when implementing this method, since it is generic to produce the embeddings from the texts of any length. This model even can be trained under an entirely unsupervised fashion from a large corpora of raw text without requiring any specific task labeled dataset. Doc2Vec has two models which are the Distributed Memory (DM) and the Distributed Bag of Words (DBOW). For the DM model, the probability of a word given in the document vector and context will be predicted. On the contrary, for the DBOW, the probability of a set of random words in the document given a document vector will be predicted. Along with these two models, the document vectors are shared during the training of a single document in order to employ the semantics of the entire document during predicting the probability of the words. For the example for a sequence of training words $w_1, w_2, w_3 \dots w_T$, the purpose of this model to escalate the average log probability:

$$\frac{1}{T} \sum_{t=k}^{t+k} \log p(w_t | w_{t-k}, \dots, w_{t+k})$$

Figure 2.16 : Average log probability of Doc2Vec

Also, the probability is calculated using the softmax function whereas is determined as the following mathematical operation:

$$p(w_t | w_{t-k}, \dots, w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}}$$

Figure 2.17 : Softmax function for probability of Doc2Vec

CHAPTER 2: LITERATURE REVIEW

2.1.5 Strength and weakness

In previous studies, there are some limitations or weaknesses on several techniques and methods. In order to further improve the proposed system or studies, the drawbacks in personality traits, sentiment analysis and natural language processing are carried out.

By looking at the personality and emotion model, the Big Five Factor model was implemented to approach the personality traits rather than as a comprehensive theory of personality. According to stated, it is more descriptive than explanatory and does not fully justify the differences between individuals. Besides, it does not accomplish a casual reason for human behavior. Due to cross-cultural validity, the most previous studies have tested the presence of Big Five in urbanized literate populations. This will cause some overall participants did not include the instances consistent with this Big Five trait model. On the other hand, OCC model depend on patterns of antecedent of emotions approach causation, attribution and eliciting circumstances conducive to determine emotions while the evolving circumstances or attribution are not correlated directly from the dimensional approach. It might fall short in publishing the reason certain feelings are elicited.

Along these factors mentioned above, the lexicon-based approach to sentiment analysis can be carried out easily and quite intuitive to use. Nonetheless, technique will be unsuccessful in the ambiguous phases due to its words in the sentences not necessarily matching the primary meaning of the words. In addition, the context of the opinion or expression does not conduct into consideration for further data analysis with this approach. Obviously, the construction of an exhaustive lexicon is an exhausting yet tedious process. Apart from that the machine learning methods are somehow easier to implement and suitable rather than lexicon-based methods. It is because while machine learning determines the polarity of the sentence, they provide different kinds of features and their relationships in various into consideration. Nevertheless, there is a drawback as a machine learning based sentiment analysis method requires a huge amount of training dataset.

CHAPTER 2: LITERATURE REVIEW

In sentiment analysis topic models, the limitation of TF-IDF is widely used due to its conciseness and can apply into account the weight of words. However, it can extract all sorts of words. Not only a specific aspect of the substances. Besides, this method cannot identify and reflect the trait of a specific aspect of the user. It also reveals little about inter-intra document's statical structure and only a relatively small amount of reduction. Rather than that, the LDA model sometimes will be correlated with other topics as Dirichlet topic distribution cannot derive its relationship and correlation. Besides, it sometimes will be weak in supervision of sentiment analysis. However, LDA is central to topic modelling and always revolutionized and developed in this field despite its limitations.

Application Programming Interface (API)

API is a list of programming codes that allows data transmission between the software or system each other which is in terms of data exchange, data communication and data interaction. This enables the services, applications and products to access and interact with each other and leverage the data and functions with the documented interface . For instance, the Instagram application, there are some features including the ability to send an instant message, check the location on maps or weather on the phone which depend on using an API.

The API is a set of defined rules that demonstrates how the applications of computers communicate with one another . APIs will take place at an application and the web server to operate as an intermediary layer that possesses the data transfer between the systems.

Based on the following diagram, a web application in browser will initiate an API call to collect the information which is called as a request. The request can proceed from an application to the web server via this API's Uniform Resource (UPI) which can involve a request verb, headers or the request body through the internet. The API also builds a call to the external program such as a database or web server after receiving the valid request. Also, the server sends a response to the API with the requested information. The data will be transferred back by the API to the particular initial requesting application. This process of the request and response will be incurred through an API

CHAPTER 2: LITERATURE REVIEW

during transferring data on the web service between the computer or an application. [21]

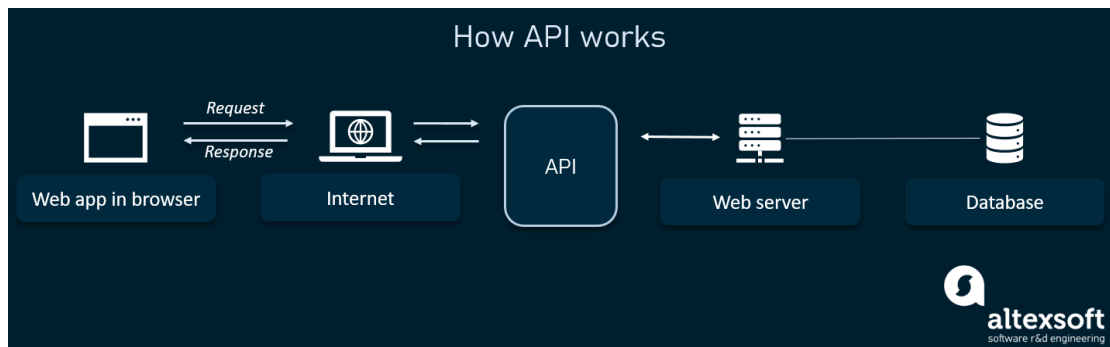


Figure 2.18: How API works

Programming Language - Python

https://www.w3schools.com/python/python_intro.asp

Python is one of the most famous yet high-level programming languages used for software development, web development for the server side, Machine Learning applications, web frameworks like Django, GUI Application like PyGT, mathematics, web scraping like Scrapy and system scripting. Compared with other programming languages, it was created for readability and more similarities to the collerative of English language and mathematics which allows the non-programmers easier understanding. It also can be used in a procedural way and object-oriented way or a functional way. Besides, it provides a more simple syntax and can work on different platforms such as Windows, Mac and Linux and so on. Thus, it is very suitable for the beginner for a wide range of purposes and development for applications. [22]

Web Frame (Backend) - Flask

[\(https://pythonbasics.org/what-is-flask-python/\)](https://pythonbasics.org/what-is-flask-python/)

Flask is a web application framework that is used in Python. It is a lightweight WSGI web application framework that is widely used in Python and designed to implement easy and quickly by scaling up with the complex applications. It also starts as an uncomplicated wrapper around Werkzeug toolkit and Jinja2 template engine.

CHAPTER 2: LITERATURE REVIEW

Werkzeug is a Web Server Gateway Interface (WSGI) toolkit to carry out the functions including the request, response objects and utility those function and set as bases. Jinja2 is a famous template engine for Python that combines a template with a specific data source to contribute to the progressive web page. Moreover, it is a beginner friendly and scalpel for newbies to set as foundation or template essentially to build the web applications or websites and services. [23]

2.2 Overview of research

2.2.1 Limitation of Previous Studies

In previous studies, there are some limitations or weaknesses on several techniques and methods. In order to further improve the proposed system or studies, the drawbacks in personality traits, sentiment analysis and natural language processing are carried out.

By looking at the personality and emotion model, the Big Five Factor model was implemented to approach the personality traits rather than as a comprehensive theory of personality. According to John & Srivastava, (1999) stated, it is more descriptive than explanatory and does not fully justify the differences between individuals. Besides, it does not accomplish a casual reason for human behavior. Due to cross-cultural validity, (McCrae, 2002) the most previous studies have tested the presence of Big Five in urbanized literate populations. This will cause some overall participants did not include the instances consistent with this Big Five trait model. On the other hand, OCC model depend on patterns of antecedent of emotions (Ortony, Clore and Collins, 1998) approach causation, attribution and eliciting circumstances conducive to determine emotions while the evolving circumstances or attribution are not correlated directly from the dimensional approach. It might fall short in publishing the reason certain feelings are elicited.

Along these factors mentioned above, the lexicon-based approach to sentiment analysis can be carried out easily and quite intuitive to use. Nonetheless, technique will be unsuccessful in the ambiguous phases due to its words in the sentences not necessarily matching the primary meaning of the words. In addition, the context of the opinion or expression does not conduct into consideration for further data analysis with this approach. Obviously, the construction of an exhaustive lexicon is an exhausting yet tedious process. Apart from that the machine learning methods are somehow easier to implement and suitable rather than lexicon-based methods. It is because while machine learning determines the polarity of the sentence, they provide different kinds of features and their relationships in various into consideration. Nevertheless, there is a drawback as a machine learning based sentiment analysis method requires a huge amount of training dataset.

CHAPTER 2: LITERATURE REVIEW

In sentiment analysis topic models, the limitation of TF-IDF is widely used due to its conciseness and can apply into account the weight of words. However, it can extract all sorts of words. Not only a specific aspect of the substances. Besides, this method cannot identify and reflect the trait of a specific aspect of the user. It also reveals little about inter-intra document's statical structure and only a relatively small amount of reduction. Rather than that, the LDA model sometimes will be correlated with other topics as Dirichlet topic distribution cannot derive its relationship and correlation. Besides, it sometimes will be weak in supervision of sentiment analysis. However, LDA is central to topic modeling and always revolutionized and developed in this field despite its limitations.

2.3 Proposed Solution

By using the psychological models, the content text from the email or messages sent can be mapped to retrieve the core emotion and intensity. This can analyse and identify which emotions or attitudes are most important expressed in the content text data. The following stated emotional model can be deployed in conjunction with the proposed systems to implement the task of emotion extraction. To tackle the drawbacks of the techniques and methods I have studied, a new model system can be proposed for sentiment analysis for detecting an individual emotion category. In this model, we aim to have some documented process by applying the suitable techniques and methods with the following steps:

1. Retrieval of Data:

The input can be any message or text content from the email or social media chat platform to ease of data extraction. The data can be mined with the existing online social communicant media such as Facebook messenger, WeChat, Instagram, twitter and WhatsApp etc. the data input will be chosen and collected based on some content that can pertain to the domain of the concerned topic related such as conversations between two friends.

2. Emotion extraction:

Personality and emotion models always mapping the core emotions to a computational scale from which I can easily classify predominantly and detect the particular personality and emotions expressed. In my purposed of our system, the consideration of the model is Big five Dimension model as it divides 5 basic personality traits based on the behaviors and emotions of an individual. This model can represent the complexity and intensity of the human feeling with collaboration of sentiment analysis. For example, the Agreeableness, Conscientiousness and Neuroticism in Big Five factor model has deployed these emotions and behaviors predicted and analyzed to detect whether people are toxic or not.

3. **Preprocessing:**

Before, the feature extraction with the classification, the sentences or the content given can be built as a feature vector and goes through the natural language processing. The following techniques and methods studied in literature review can be performed to convert the data input of the messages or email into processable elements with more details for further utilization in feature extraction. Tokenization is the process to convert the plain context as a string into process labeled elements, namely tokens. Besides, Parts of Speech (POS) tags are extracted to handle out-of-vocabulary words (OOV) to identify the characteristic of a word in a sentence according to grammatical categories of the words of a language. These measures can remove some 'noisy' and unnecessary data through a series of filtering and mapping processes.

4. **Text feature extraction:**

There are several sentiment analysis models being explored in this study. As the consideration that emails and long text content have a mixture of topics whereas a probability distribution of topic over words will be carried out. s (Blei, et al., 2003). The LDA model will be possessed in our proposed system for feature extraction and feature selection. It is because LDA is convenient to enforce as a module in more complex models. Also, it is a simple and efficient model for dimensionality reduction and summarization of the input data rather than other models. The training set of the positive and negative sentences will be divided by using LDA in order to search the suitable latent topics and contributing keywords to the topics.

5. **Text Classification:**

Typically, there are hundreds and thousands of messages or emails of an individual, thus sentiment analysis model carried out into classification

techniques. It can accomplish the large corpus of data context given and search a suitable yet effective pattern to assume and predict the personality and emotion of an individual. For classification methods, Hybrid method with the combination of lexicon and machine learning such as Naïve Bayes are being deployed in the proposed system. It is because this hybrid method has carried out high performance and high accuracy rather than two methods deployed independently. Besides, it establishes the polarity of every sentence in action handling the use of emotions into considerations. Its overall orientation of the emotions enforced in every sentence is similar to the overall sentiment expressed by the words of that plain text. Furthermore, the respective method of sentiment analysis classifier can complement each other by taking advantage to recover their weakness. Also, it can be less over-lifting and more robust to noise of the data input.

CHAPTER 3: PROPOSED SYSTEM METHOD/APPROACH

3.1 Design Specification

3.1.1 Methodology

The preferable and suitable methodology for the development of this project, Toxic Friend Detector is agile development methodology. The agile method will undergo the sprints of project planning, execution, adapting the scope and design throughout the project. Naturally, the main reason for the agile approach is that the tolerance of the changes to adapt the requirements is better than other methodologies applied. This agile management allows the ability to comply with the changes of the project. This can give the chance for this proposed project if the needs and scopes have to do some correction or further improvement. The risk will be lower due to the proposed project developing many versions regularly after discussion and getting some feedback. This can minimize the risk of a project to be unsuccessful or fail to deliver the outcomes. In addition, this project can separate into some iterations to search the small issues early and address them immediately rather than determining the large problem while at the end of outcomes. Agile methodology also integrated a continuous development approach to ease the project to continuously deliver new products. Moreover, it supports the innovation and ongoing collaboration to have further improvement on the proposed project. Thus, there is FYP1 and FYP2 for carrying out some updated versions of the project at much shorter intervals. Through several discussions and meetings, the project with agile will bring a better-quality outcome at the end.

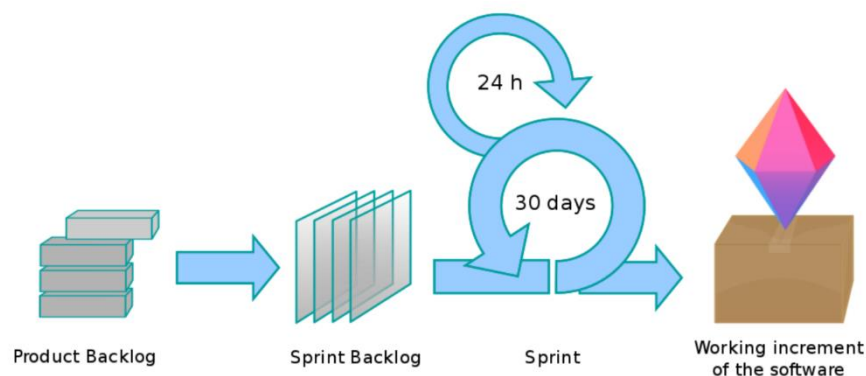


Figure 3.1: Agile Development Methodology

3.1.2 System Design Diagram

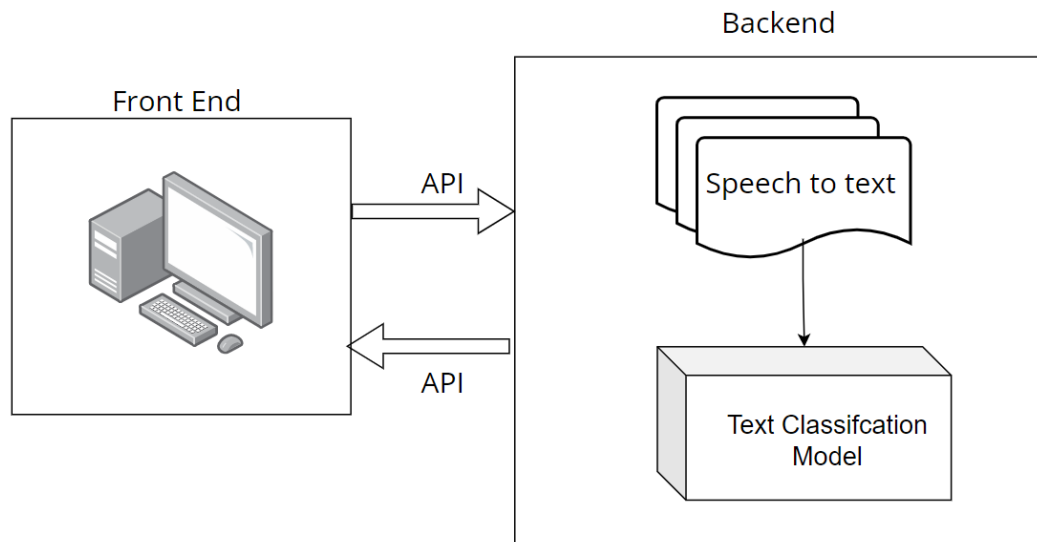


Figure 3.2: System Development Methodology

This chapter will intensify on describing the general work procedure for this development project. This project focuses on profiling the front end part which is a web page for deployment and back end part which for speech to text model and Text Classification model. For the front end part, when the users input the speech or an audio file of an individual at the webpage, the web application will initiate the API call to collect the data. It will request from this Toxic Friend Detector application to the web server via the URI. Then the API creates a call to our back end model after retrieving a valid request. At the back end, there are two parts of processes that need to be done which is Speech to text process and Text classification process.

First and foremost, the speech input will be identified and transformed into text form for further text classification in another process. The large vocabulary speaker independently disengaged chiefly in the worldwide and it is also an interaction language spoken and written widely currently. [24] Thus, there is an extraordinary need for the Automatic Speech Recognition (ASR) speaker to be conducted for people speaking the English language. This project work is implemented of the speaker independent confined speech employing Sphinx4 which is acceding on the Hidden Markov Models (HMMs) or using the Google Speech Recognition, an existing famous Speech Recognition system. There are some inputs and data downloaded from the Internet and the recording from different kinds of people. The data will be processed by some feature

CHAPTER 3: PROPOSED METHOD/APPROACH

extractions to convert the speak waveform into the model readable format. In order to conduct the speech recognition, few models were proposed to model the audio and visual modality for predicting the text during the post-pre-processing process. The decoding functions such as Acoustic Model, Pronunciation Dictionary and Language model to work together and evaluate the performance analysis. After that, the hypothesis of the system application will transcript the speech to the text file along with some available details.

After that, the text classification model has received the transcription of the speech of the text file, the text file will be determined as testing data. The testing data will proceed the initial cleaning in order to discharge some textual elements that are not be considered during the speech to text analysis. After that, there are some preprocessing text steps in the natural language processing pipeline such as tokenization text, stop word removal and stemming to reconstruct the text into a list of words which are known as tokens. The words also transformed into numbers, where more precisely an array of numbers. Before feature extraction, several visualizations are worked on the data to have a better understanding of them and categorized into the nimary labels for toxic and non-toxic. Different kinds of embeddings such as TF-IDF, Word2Vec and Doc2Vec are implemented during the feature extraction for transforming the array of the words into an array of numbers with the Vector Space Model (SVM). Encoding the stemmed text or vector of words into VSM is not a trivial approach, including several domain-based reasoning. The data will normalize the data or samples individually to unit form in the feature normalization section so that it can be sustained as input to the classification model. [25]

The main core process of the Linear SVC baseline, logistic regression baseline and Naive bayes Logistic baseline model have been tested with different kinds of encoding techniques which are mentioned before such as TF-IDF, Word2Vec and Doc2Vec for picking up the best model amongst these 2 according their performance. At the last, the logistic regression algorithm with Doc2Vec is determined as better encoding and model amongst the classifier models. The test data will be deployed with this encoding technique model and having the prediction to the label class of toxic and non-toxic. Along with the piepie;line and trained models, the web application with a graphical user interface (GUI) is developed and designed which is the front end part mentioned in the figure following. So the system sends a response to the API with the requested

information. The API will transfer the data which is the output that mentioned the individual is toxic or non toxic to the front end web application. After having some process of retrieving data, data filtering, preprocessing and carrying out the classification model, we aim to have an effect and outstanding in order to determine and evaluate the effectiveness of the proposed system. Several measurements and techniques such as confusion matrix are deployed to train our proposed model and discover the underlying performance in terms of how accurate and precise the model can reach.

3.1.3 System Performance Measurement

Confusion Matrix

Confusion matrix helps us to figure out how the model performed and check where it is wrong in the way and navigate to set up the correct path. In a confusion matrix, every entry will stand for the number of predictions created by the model where it analyzed and classified the classes correct or incorrect. Its matrix denoted the actual class versus the predicted result. The samples will be classified into four types in the confusion matrix which is mentioned in the following below.

True Positive (TP) denotes the number of positive samples that are predicted correctly as positive.

False Positive (FP) denotes the number of negative samples that are predicted correctly as negative.

True Negatives (TN) denote the number of negative samples that are predicted correctly as positive.

False Negatives (FN) denote the number of positive samples that are predicted correctly as negative.

Confusion matrix is a $N \times N$ matrix evaluated for determining the performance of a classification model, where N is the number of targeted classes. However, if we have set up several labeled classes, hence the confusion matrix for multi-class classification is executed.

		True Class		
		No_entry	Stop	80_speed
Predicted Class	No_entry	7	8	9
	Stop	1	1	3
	80_speed	3	2	1

Figure 3.3: Example of Confusion Matrix

It might be several different for the binary matrix classification, since every individual labeled class will be directed to search for TP, TN, FP and FN. For instance, the multi class classification of calculation for the No-entry will be $TP = 7$; $TN = (2+3+2+1) = 8$; $FP = (8+9) = 17$; $FN = (1+3) = 4$.

Accuracy Score

The accuracy score is the ratio of the correct prediction to the total predictions deployed. Accuracy score is implemented to analyze and measure the performance of a classifier. It provides the overall accuracy of the proposed model, which denotes that the set of labels predicted for a sample matches the corresponding set of labels in y_true . Here are following of the formula of accuracy from the confusion matrix:

$$\text{Accuracy score} = (TN + TP) / (TN + TP + FP + FN)$$

Precision, Recall and F1 score

Precision can be named as positive predictive value (PPV). Precision is the number of correct positive results that are divided by the number of positive results they analyzed by the classifier. The formula of precision will be constructed as shown below:

$$\text{Precision} = \text{TP} / (\text{FP} + \text{TP})$$

In addition, recall is denoted as sensitivity or true positive rate (TPR). Recall is the number of correct results is divided by the number of all relevance positive samples that predicted correctly. The formula of recall will be constructed as shown below:

$$\text{Recall} = \text{TP} / (\text{FN} + \text{TP})$$

F1 score is the Harmonic Mean that has the relationship between precision and recall. It is executed to measure and identify the test's accuracy. High precision but low recall will provide an outstanding actual result but will lose a huge number of instances that are difficult to recognize and classify. Therefore, the greater the F1 score, the better for the effectiveness and performance of the proposed model. The formula of F1 score will be constructed as shown below:

$$\text{F1} = 2 / (1/\text{precision} + 1/\text{recall}) = 2 \times (\text{precision} \times \text{recall}) / (\text{precision} + \text{recall}) = \text{TP} / (\text{TP} + (\text{FN} + \text{FP}) / 2)$$

3.3 Timeline

Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Upcoming Semester
Reviewing IIPSW report															
Exploring Related Work Techniques															
Planning Schedule															
Organising Report Details															
Installation Required Software															
Identify and Study the Modules															
Dataset Collection															
Data Preprocessing															
System Design															
Building Model															
Develop and Testing															
Performance Evaluation															
Finalizing Report															
Submitting FYP1															
Presentation and Demo															
Further Research															

Figure 3.4 The Gantt Chart of the proposed project development

The estimated timeline of this project for deliverables and milestones is at xx 2022. For Final Year Project 1, has completed within one trimester which has mainly focusses in exploring the former works, preparing data and developing the model. During the development of the system, block diagram, system diagram, Gantt chart, and some paperwork are conducted and run consecutively to completed. Before the end of this trimester, the model will be implemented and testing for more realistic speech recognition and outcome a better result. Furthermore, the presentation about this project and demo of this FYP1 will be done to supervisor and moderator between week 12 and week 14.

CHAPTER 4: SYSTEM DESIGN

4.1 Project Flow Diagram

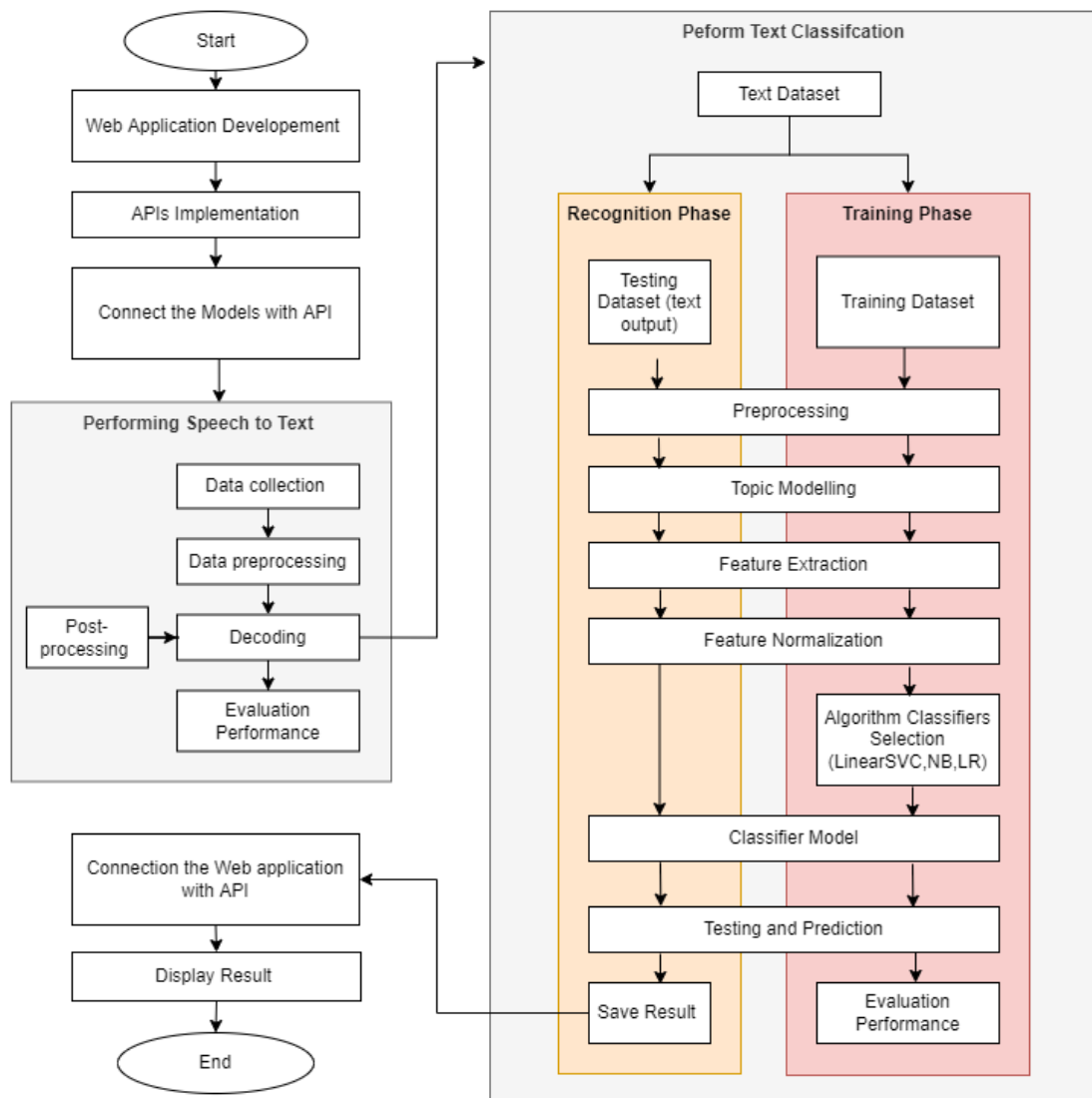


Figure 4.1: Project Flow Diagram

The inception of the project is to identify an individual whether toxic or not. Thus the speech or conversation of an individual will be collected and then will use our proposed toxic Friend Detector system via web application to detect the personality of an individual. In this section, a detailed flow diagram for overview system architecture is being illustrated to explain every model or any design with the meticulous description. There are two main parts which are Frontend and Backend parts. The front end part will involve the web application flow and API flow, while the Backend part will involve the speech-to-text model and text-classification model. The first stage of the Front end part

is developing web applications with a GUI. The web development will use HTML, Javascript and CSS to create a simple web application for the users to use it. The API with the Flask library will be implemented to indicate the back end of python models which are the speech to text recognition model and the text classification model.

After successfully sending the API request, the speech model of the sphinx4 will be performed to retrieve the speech file and transcript the speech to text. In this backend part of the speech-of-text model, the audio file will undergo the data preprocessing with some features extraction techniques such as the Mel-Frequency Cepstral Coefficients (MFCC) , bark frequency wrapping and so on. While the linguist module with three main formed components including the Acoustic Model and the pronunciation Dictionary at the post process will be carried out to further encoding in the next steps. [26] During the decoding of the speech, the search manager in the decoder block with the help of a search graph will proceed to some features and perform the search algorithms to pass the tokenization. After the speech file being recognized and transcript successfully, the output result file will be sent to another model which is Text classification to further classify the toxic and nontoxic labels. This process can be done offline. On the contrary, some of the built-in features such as Watson developer cloud and Speech recognition, whereas Google Speech-to-Text can identify the user's intent to transcript the speech to text with online and more flexibility in any Python project. Thus there is the performance of the speech recognition model being evaluated to identify which one is better for further classification.

Subsequently the following of the second stage in backend part which is the Text-classification is well performed by carrying out several steps that are similar to the speech to text model. The data collection will be retrieved from the internet and the data preparation will retrieve the given speech-to-text file from the speech recognition model. The text classification model will be constructed in 2 phases which is filtering out one for training phase and recognition phase. These two phases will be carried out on the data preprocessing, topic modeling for proceed the feature extraction and feature normalization. Also, during the training, the topic modeling will be embedded with the classifiers such as Linear SVC, Logistic Regression (LR) and the Naives bayes to fine tuning which one model has a better accuracy and performance. After having the better classifier and the pipeline, the identification and testing can proceed to categorize the text file in the class of toxic or non-toxic labeled.

Continuing with one of the stages of the front end part, the API will trigger the defined model to get the result response from the text classification model. API will call and connect with our Toxic Friend Detector web application system in order to display the results of the toxicity of an individual.

4.2 Web Application Flow

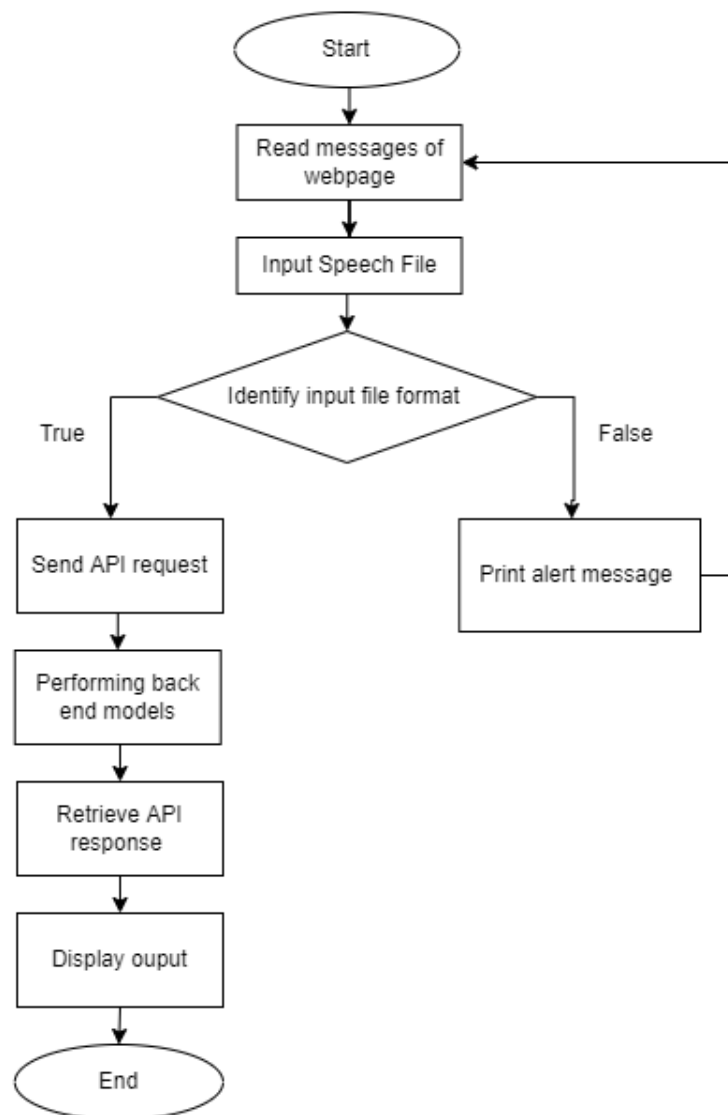


Figure 4.2: Webpage Block Diagram

The figure above illustrates the webpage Block Diagram. The webpage is designed by Javascript, HTML and CSS to provide a platform for the users interacting with the Toxic Friend Detector system. It is constructed with simple GUI design and one feature for users to approach only. The user can read the message and content of the webpage

to understand what file needs to be inserted in the system. When the user uploads the audio or speech file with correct format, the system will continue to call the API. The backend models will continue performing its tasks and respond back the result to this web application. The web application will retrieve the data and display the result that an individual is whether toxic or nontoxic according to the speech file. If the wrong format audio and speech file is uploaded, the system will prompt an error message to alert the user to reupload the correct file.

4.3 Speech-To-Text Procedures Flow

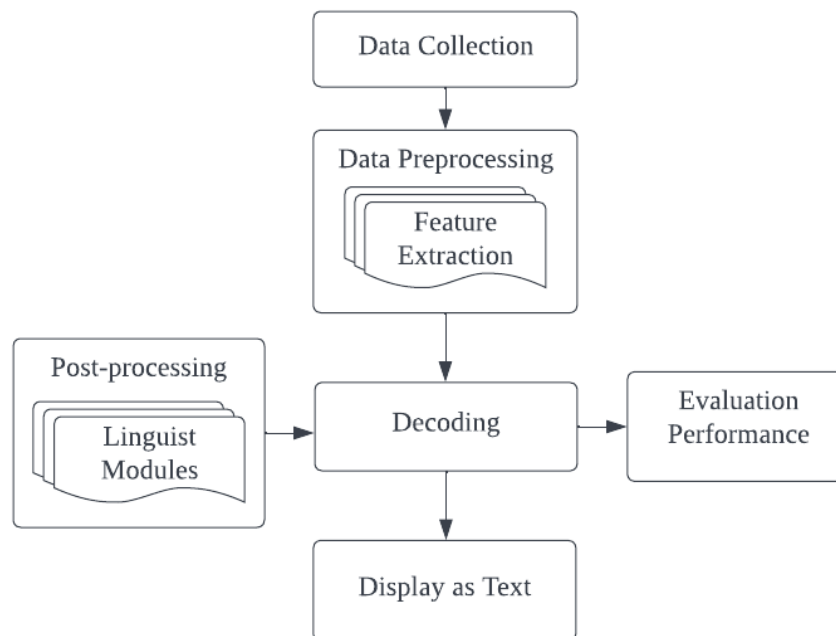


Figure 4.3: Simple Speech-To-Text Flow Diagram

This chapter will intensify on describing the general work procedure for this development research. This project focuses on profiling the speech to text of the speech recognition of an individual. The large vocabulary speaker independently disengaged chiefly in the worldwide and it is also an interaction language spoken and written widely currently. Thus, there is an extraordinary need for the [27] Automatic Speech Recognition (ASR) speaker to be conducted for people speaking the English language. This project work is implemented of the speaker independent confined speech employing Sphinx4 which is acceding on the Hidden Markov Models (HMMs).

During the early phases of this project methodology, there are some inputs and data downloaded from the Internet and the recording from different kinds of people. The data will be processed by some feature extractions to convert the speak waveform into the model readable format. In order to conduct the speech recognition, few models were proposed to model the audio and visual modality for predicting the text during the post-pre-processing process. The decoding functions such as Acoustic Model, Pronunciation Dictionary and Language model to work together and evaluate the performance analysis. Lastly, the hypothesis of the system application will transcript the speech to the text file along with some available details.

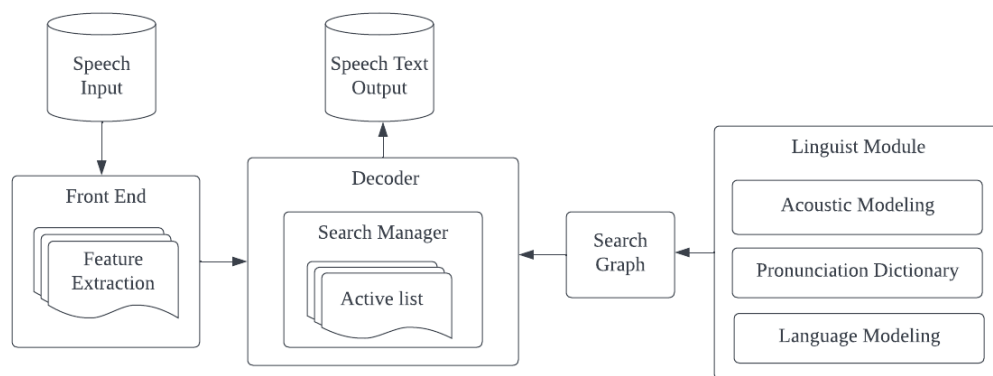


Figure 4.4: Block Diagram for Speech-To-Text

4.3.1 Data Collection

The data will be obtained from any speech material for training, testing and optimizing the speech recognition system. The data must be in audio form since need to transcript the speech to text. The audio data can be collected from the internet resources, and it is not important to be large file size. For example, the movies, audio books, podcasts that support constructing a better good acoustic model with some effort. The speech file also can be a couple of recordings of yourself. However, the audio material must specify the language that is going to be used in the model. The speech data will be downloaded onto the host machine first before proceeding to the next process. [28]

4.3.2 Data Pre-processing

Since the primary adoption to data preprocessing or front-end speech recognizer is to compose speech waveform to the extraction of the speech. This is a process of feature analyses to depiction of the signals investigated to construct a variety of predictions. The Front End resides to have one or many parallel chains of convertible communicating signal processing modules which are attempted as Data Processors. Data Processors will implement and concert speech waveform by extracting some techniques such as the following: [28]

- (i) Reading from a range of input formats for assortment mode operation
- (ii) Interpretation from the system audio input device for live mode application
- (iii) Preemphasis
- (iv) Windowing alongside a raised cosine transform for example the Hamming and Hanning windows
- (v) Discrete Fourier transform (via DFT)
- (vi) Mel-frequency filtering
- (vii) Bark frequency warping
- (viii) Discrete Cosine transform (DCT)
- (ix) Mel-Frequency Cepstral Coefficients (MFCC) method
- (x) Cepstral mean normalization (CMN)
- (xi) Linear predictive encoding (LCP)
- (xii) End pointing
- (xiii) Perceptual linear prediction coefficient extraction (PLP)

From these feature extraction methods, the Mel-Frequency Cepstral Coefficients (MFCC) is the most indisputable instance of feature sets that is widely conducted in

speech recognition. For instance, with the following steps that included in MFCC feature extraction that will be employed for further of post processing: [29]

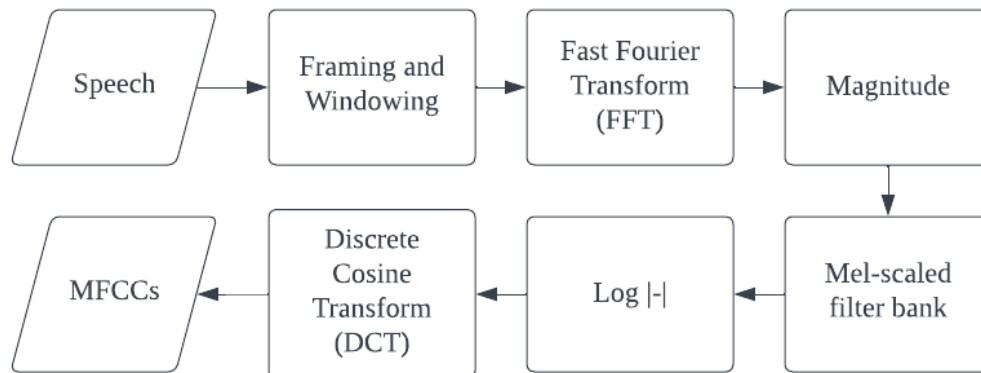


Figure 4.5: MFCC Block Diagram

The pre-emphasis will strengthen the energy and support in high frequencies to provide more information to the Acoustic Model so that enhancing the performance of speech recognition. To introduce the least amount of distortion for signal, the windowing and framing will be extracted once every 10ms which is set as frame rate. The coefficients on the Mel scale that perform sounds while the words cepstral approach form the word cestrums which is under the spectrum's logarithmic scale. The technique of computing for the Fast Fourier transform is to come on performing each frame of speech data and the magnitude and then conduct the frequency-wrapped set of log filter banks which are Mel-scale filter banks to filter the signal.

The logarithmic scale will be set as a measure of perceived frequency or pitch of a tone, alongside the log filter bank amplitudes. Human ears are less sensitive so it can stimulate the hearing scale of individuals. Lastly, construct the Discrete cosine transform (DCT) to calculate MFCCs and some additional sets of delts and acceleration coefficient features. The first and second-time derivatives of the original will be extracted respectively. According to the short-term analysis each frame of MFCC vector is computed and calculated by conducting this following equation:

$$Mel(f) = 2595 \log\left(1 + \frac{f}{700}\right)$$

With generic of the Front-end Framework, these Data processes are subjected to conducting a data object conceived of parameterized signals which identifies as features which to be constructed by the decoder. [30]

4.3.3 Data Post-processing

Linguist Model

The Linguist model is made up of the three constituent pluggable components: The Acoustic Model, the Pronunciation Dictionary and the Language Model which have to be implemented to gather during the recognition process. Also, these post-processing components are proposed to carry out the Search Graph that will be constructed by the decoder during the search and concealing the complexities included during the same time of implantation this graph. [31] [32]

Acoustic Model

This component of the linguist is characterizing sounds of the languages, also the diversity of feature vectors of the collection files. The plan of acoustic modelling is to generate robust statistical models and construct acoustic properties for each senone. It is built from the context-dependent, which is from senones with context, also from the context-independent modules that apparently feature vectors for each phone. Hence the file dictionary, binary language module, transcription, filler, phone and file aids can be sets as required data of this model as speech training databases.

These fundamental speech units will be extracted as probabilities of a trained recording into acoustic models which it's based on the Hidden Markov Model. (HMM). The use of HMM structural mechanism for modeling speech statistics and dynamics to regulate the hidden parameters from some observable parameter from this project assumption. Obviously, the association of the HMMs and acoustic phonetic conducts to detect and correct linguistic irregularities and strengthen the robustness for recognition to the input audio in noise.

Pronunciation Dictionary

The Pronunciation Dictionary can also be served as a Phonetic Lexicon that includes the information and details about the pronunciation of the words. It also works as a mapping of the words which is grapheme into phoneme sequences (phones). The pronunciations will be fractured and break into the sequences of the sub word units in the Acoustic Model. The occurrence of every word for sentence and transcripts or spoken corpus will be deployed in the dictionary. The sample of pronunciation dictionary for English speech are stated as following:

abandon	AH B AE N D AH N
gravity	G R AE V AH T IY
organization	AO R G AH N AH Z EY SH AH N
worksheets	W ER K SH IY T S

Table 3.2: Audio Format Settings

Language Model

The Language Model is a set of probabilistic data formed from the sequence of transcript words [33]. This form of probability can be a unigram or N-gram model. The unigram model is usually used in information retrieval. The N-gram model is used to estimate the length of word phrases or sentences and sequences which are not observed during training of the model. The Language Model is a set of probabilistic data formed from the sequence of transcript words. This form of probability can be a unigram or N-gram model. The unigram model is usually used in information retrieval. The N-gram model is used to estimate the length of word phrases or sentences and sequences which are not observed during training of the model. The language model is implemented to restrict word search served as word-level language structure. It supports the module checking which word could select in accordance with the precious recognized words and side-finding the matching process by stripping the probable words. It also represents the expectation at utterance and former knowledge about the language. The rules are conducted at this stage to express the linguistic restrictions that define the language and to allow the reduction of possible unidentified phoneme sequences. Commonly the language models are formed as unigram, bigram, trigram, or N-gram so

that can be set up in information retrieval and then estimation of the word phrases/sentences length during the training model.

Search Graph

As declared before the linguist conducted its three components to generate the SearchGraph, [28] to conduct in the decoding process. It is defined as a class of algorithms that compute the properties of parameters and also a data structure by deployed affects the speed, memory footprint and recognition accuracy. As an example of the directed graph any of a node which is served as a Search State that defines an emitting or anon-emitting state. Generally, the emitting states can be achieved against incoming acoustic features in contrast to the non-emitting states deployed to serve as higher lever linguist constructs such as phonemes and words that are not scored straightforwardly against the incoming features.

4.3.4 Decoding

The fundamental role of the Sphinx4 Decode block is to conduct the features extracted by the Front-End and the Search Graph from the Linguist. The elementary component of the Decoder block is the Search Manager which can recognize a set of feature frames and perform search algorithms like frame synchronous. Each Search Manager construction implements a token passing algorithm as defined with a Search State and the speech path of the given point. Every partial hypothesis will terminate in an active token. Besides, the Search Manager also will get back the interim results as the proceeds of the speech recognition and also the final results after compiled recognition. The sub frameworks of Search Manger will generate and composed of an Active List, a Pruner and Scorer. [33]

The Active List is determined from the former active tokens in search trellis by pruning and deploying a pluggable pruner. Application can configure the Sphinx-4 conduction of the Pruner to deploy both relative and absolute beam pruning. Pruner also can act as garbage collector of the Java platform so that it can prune a complete path by slightly discharging the terminal token of the path form ActiveList. In order to permit the garbage collector to reclaim the associated memory, the removal of terminal token will

determine the token and unshared tokens for the particular path as unused. In addition, through the communication with Scorer, a pluggable state estimation module that supports state output density values on request. While the Search Manager applies a score for a given state at the particular time, the Scorer approaches the feature vector for that certain time and implements the mathematical operations to compute the score. At the same time the Scorer maintains all information related to the state output densities.

4.3.5 Display Result

While the speech file is successfully classified and recognized, the text output will be shown beside the hypothesis in the compile window. From here, this can reveal that the text has been successfully matched and recognized. Contrarily, if unsuccessfully, the broken and strange text output will be detected in the text classification model and may indicate a bad result. Besides, speech to text result for the whole transcription will be displayed at the web application after text classification process for the toxicity of an individual.

4.4 Natural Language Processing and Machine Learning Trained Model

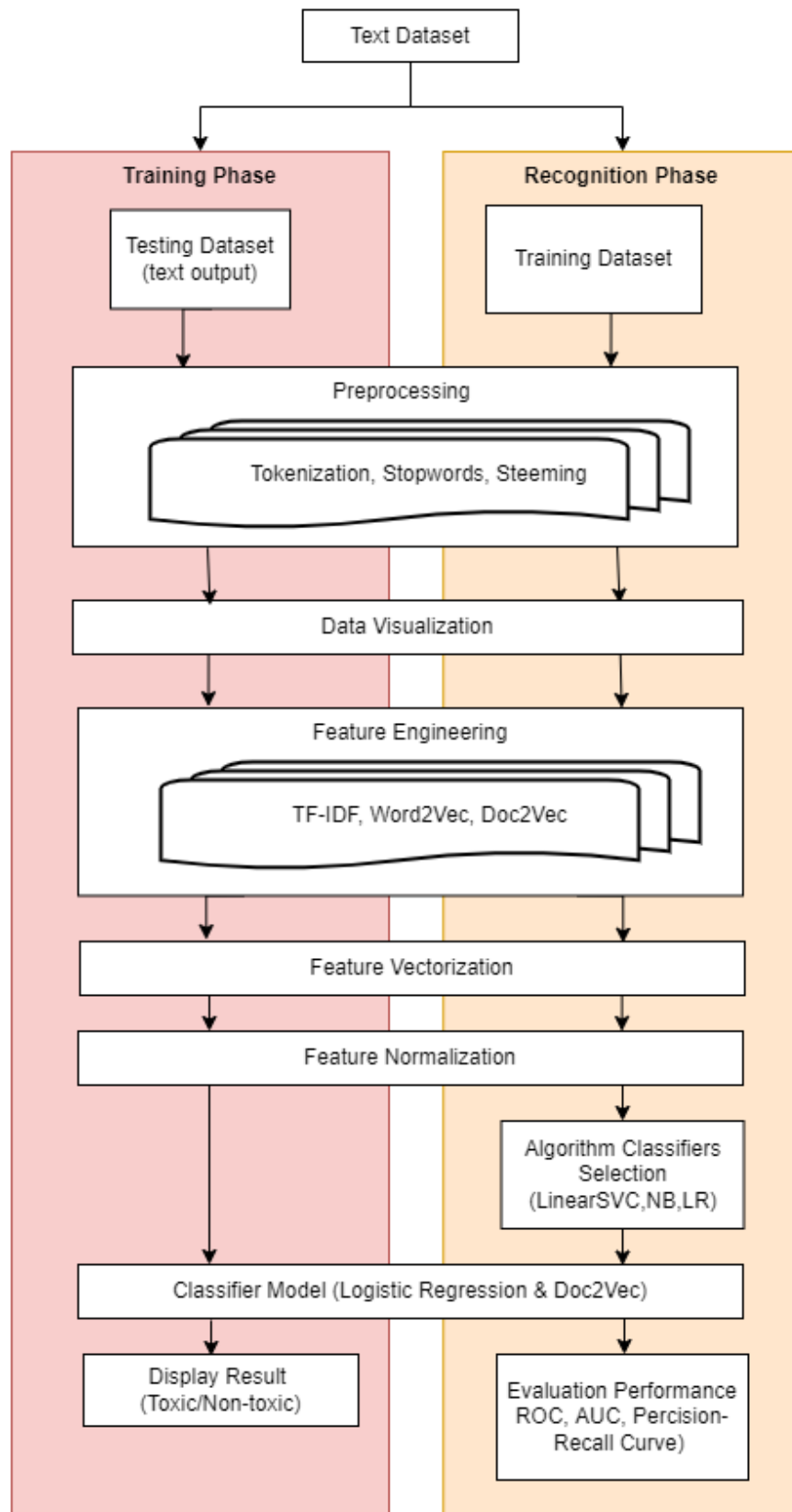


Figure 4.6: Text Classification Model Block Diagram

CHAPTER 4: SYSTEM DESIGN

The following diagram has demonstrated the flow of the Text Classification Model Block Diagram with two phases which is the Training phase and the recognition phase.

4.4.1 Dataset collection and preparation

The training dataset has been provided by Kaggle which is one of the AI competitions that focus on the research on the negative and positive behaviors such as the toxic and non-toxic comments etc. Thus, the train set is deployed and split into train and test sets commonly. The dataset involved 16225 toxic comments and 143346 non-toxic comments which were assigned manually according to the descriptor of the owner of the dataset. While the real test data is collected from the real-environment individuals. The data collected is audio or speech format during individual conversation etc. and then the test data will undergo the process speech to text recognition and output a text file. The text file will be obtained as our test dataset together with the training set.

4.4.2 Data Pre-processing

In the initial process, the text preprocessing will be carried out through the NLP pipeline to discharge some stop words and tokenize the text and also stemming the tokens. This can preprocess the unstructured text and further extraction.

Tokenization

Tokenization is also known as the word segmentation. This technique can break up the text proposed into smaller units which are called tokens. The tokens can be determined as numbers, words or punctuation marks which are also the building blocks of natural language. It can be constructed in various approaches, for instance locating the word boundaries, isolating at the level of words, character or subwords. Thus the starting and ending point of the next token calle the word boundaries.

Stop word removal

Stop word removal; is one of the general processes in NLP. The text data that is not so significant and useful for the data extraction and decision constructing like pronouns and conjunctions, articles and prepositions are categorized as non-semantic words. These words are commonly illustrated as stop words in these preprocessing steps. needless to say that those words will be discharged from the text data as a result of their minimum and lack of contribution in the information sharing about the test sentence sentiments.[34] Some of the stops words in English language like the pronouns or verbs of the “hmmm”, “arhh” and “yoo” which can affect the misunderstand and confusion during the text classification process. By applying this technique on the text data in the feature vector, the result can be discovered better.

Stemming

Stemming is the technique to reduce the inflected words into their root word in order to normalize text . It is because the same words with different injection can be redundant in the NLP process. Thus, it is also determined as a crude heuristic process that cuts off the ends of words to reach the goal effectively and correctly in the meantime and involves the removal of derivational affixes.

4.4.3 Data Visualization

This technique can let people easier to understand and interact with the data. The visualizations in this project will be indicated with the word cloud of the non-toxic and toxic raw comments which are retrieved from the wordcloud library. In order to simplify text analysis, the word cloud is a visual display of text data. The word clouds illustrate the most common or prominent words in the text bodies.

4.4.4 Feature Engineering

In order to encode the stemmed text which is also called vector words from the train and test sets into the Vector Space Model (VSM), the array of words needs to be converted into an array of numbers. It is because the processed data can be deployed in the machine learning algorithm for the classification task.

4.4.5 Feature Vectorization

There are some different kinds of methods that can be implemented for vectorization of the text such as TF-IDF Vectorizer, Word2Vec and Doc2Vec.

TF-IDF Vectorizer

Term Frequency-Inverse Document Frequency (TF-IDF) is quite famous technique performed to deploy the words significant across all the documents in the corpus. In order to convert the text data into the numeral form, the tf-idf vectorizer is deployed to convert a variety of raw documents to a format of matrix TF-idf features.

Prediction in general about a word occurs many times in the document but does not exist in other documents which means that the word is significant in that particular document. Thus this technique will apply a weight to each word according to the occurrence frequency. On the other hand, less words exist in the particular documents which mean it might bring less data for the particular document mentioned. Thus, the set of words such as a comment will be represented with an array involving the scores of the words. This technique is also quite related to the method “Bag of Words” which can give several difficulties in respect to better methods such as losing the word order and not considering the context.

Word2Vec

This technique is implemented for handling the neural network in order to learn the word representation in front of the large corpus of text. The highlights for this algorithm is its self-learned semantic representation of the words, which has quite similar function in the Vector Space Model. We use Word2Vec for implement a

continuous bag-of-words (CBOW) representation. By following the techniques of the particular words is identified from a window of contextual words. This support the average of the representation of the words that create the document (comment) to apply a representation of a document (a comment).

Doc2Vec

Doc2Vec is the creator model and performed as a persistence store which is totally different to Word2Vec. It is employed to implement a vector representation of a group of words that is analyzed as a distinct unit. By using this method, we can represent a document (comment) into a vector directly. It is an advancement of Word2Vec but adds on the unit named Paragraph Vector. There are two models deployed in this technique which is the Distributed Bag-of-Words (DBOW) to guess the content words from a certain word, and the Distributed Memory (DM) model to guess a single word from the context of words. In this project system design, the DBOW model has been approached to execute the technique.

4.4.6 Feature Normalization

The data normalization by deploying the normalizer from sklearn. can normalize the specimens or the samples to unit form individually. Every specimen or sample can be rescaled independently because of its norm (l1, l2 or inf) which defines one. The scaling method is the general operation for text classification to construct the normalization for every technique deployed to encode the text data. This brings the normalized vectors to be set as input for the classification models and even enhance the final performance of the model. The embedding techniques such as TF-IDF normalization, Word2Vec Normalization and Doc2Vec Normalization are carried out in this process.

4.4.7 Algorithm Classifiers

There are different kinds of baseline models will be contested and tested in those following methods deployed such a Linear SCV baseline, Logistic Regression

baseline and Naives Bayes baseline will embedding with the encoding techniques which isn TF-IDF, Word2Vec and Doc2Vec for further selecting the best algorithm and perform the hyperparameter tuning. In order to evaluate the performance with these different models and techniques, one of the custom functions is deployed to return a dataframe with sme evaluation metrics by the following performance measurement that has been mentioned in Chapter 3. One of the models will be identified as the best in encoding and model amongst tem. Besides, the hyperparameter search will be carried out to find the best parameter with the method of Randomized Search.

4.4.8 Evaluation Performance

In order to understand how the model performed well or not, this evaluation process is significant to construct two custom functions. Firstly is converting the probability belonging to a class into a final binary class by supporting a specified threshold. Moreover, the data frame has to be returned with the following metrics for the evaluation of the model. A custom function will take place for converting the probability to a class by implementing a threshold t. Moreover, the ROC, AUC and precision-recall curve will be performed in order to construct the “:trade off” between all the metrics. A/sl, the threshold tuning is constructed as it is a significant metric to maximize the precision versus the recall according to the domain and the application by its own.

4.4.9 Display Result

The text file is classified and recognized successfully, the output of threshold and probability and the score will be shown. The result of the toxic or non-toxic will be based on the threshold with combining the score for the text dataset. The output result will be retrieved and send an API response for connecting the web application with GUI. On the contrary , if unsuccessfully, no output will be displayed and the API will call the error message to the front end.

4.5 API Flow

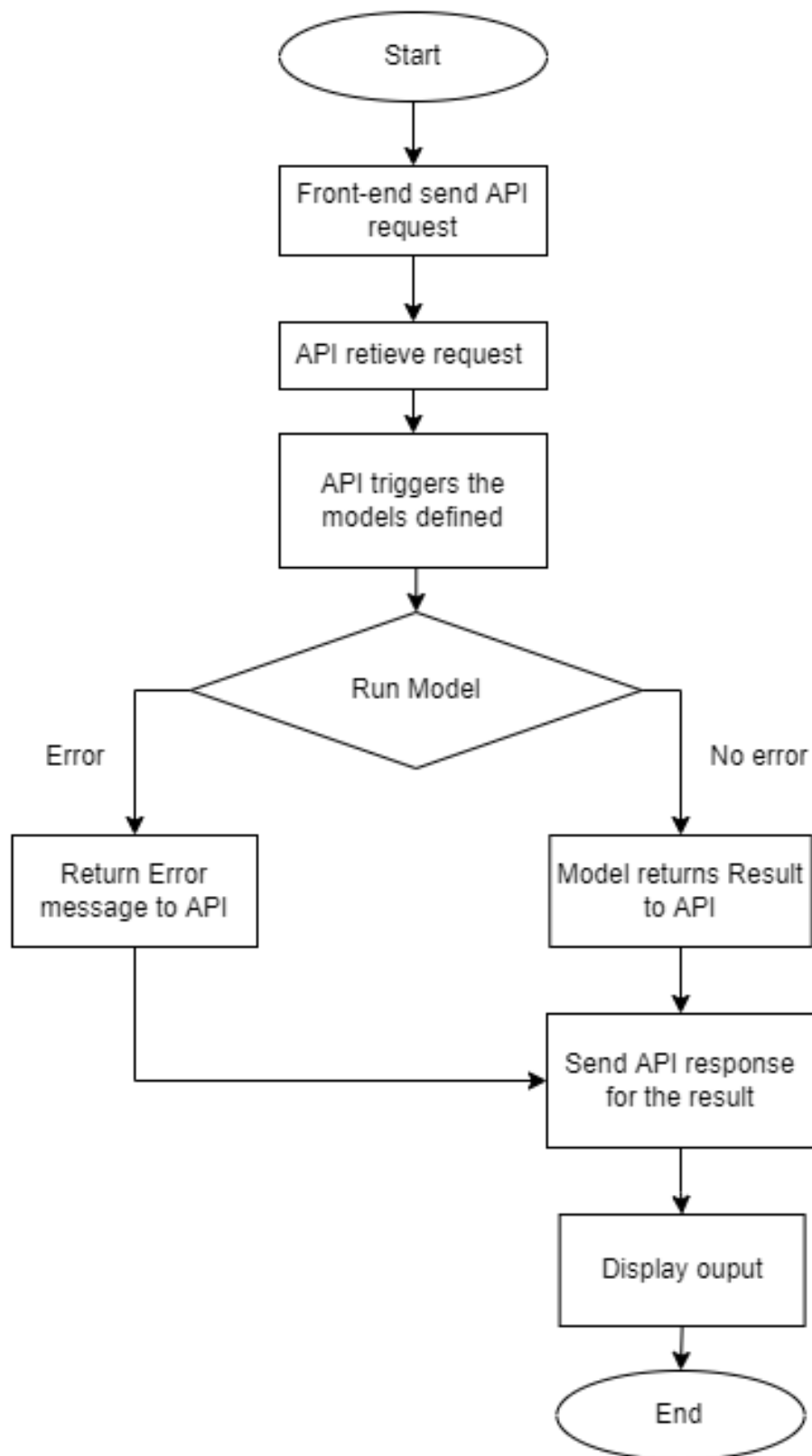


Figure 4.7: Overall API Block Diagram

CHAPTER 4: SYSTEM DESIGN

The following overall API Block Diagram has shown that the API will begin to retrieve the request once the front-end web application has sent the request. It is developed in the form of a request and response under the JavaScript Object Notation (JSON) format to query the particular system or server for the specific data and information. It will trigger the model defined and send the request data to the respective models. When the models run without any error then the particular models will continue the process and address the result to API back. The API will get the response and prompt to the web application proposed in order to return the results. However, if the error existed or failed to connect, the error message will be prompted to notify and send the error message details back to the web application proposed too.

CHAPTER 5: SYSTEM IMPLEMENTATION

5.1 Hardware Setup

The hardware involved in this project is a laptop to implement the system as stated:

Description	Specifications
Model	HP Pavilion Gaming Laptop 15-dk0xx
Processor	Intel Core i7-9750H
Operating System	Windows 10
Graphic	NVIDIA GeForce GTX 1660Ti
Memory	16GB DDR4 RAM
Storage	1TB SATA HDD

Table 5.1: Specifications of laptop

5.2 Software Setup

The software that involved in this project to implement the system as stated:

i. Linguistic Inquiry and Word Count

In order to detect the feeling polarity in positive and negative effects of a data input is based on emotions. LIWC is deployed to analyze the structural, cognitive, and emotional components and also positive and negative of the text content according to the dictionary that contains words and the classified categories. For instance, the word “allow” can be suited to the categories: affective, positive emotion, positive mood, assent and cognitive process. These sentiment analysis tools will be used in association with other methods and methods for implementing a training dataset in supervised machine learning techniques.

ii. Java SE Development Kit 1.8

Java programming language can be developing and testing in the programs through this JDK development environment. Its API supports the core functionality of the programming language and illustrate the basic types and

objects to the core API for development tolls, deployment virtual machines, and other class libraries and toolkits etc.

iii. Eclipse IDE

Eclipse is an open-source project of Eclipse Foundation and is a extensive stretchable set of equipment for software development. The Eclipse's Integrated Development Environment (IDE) component is also implemented to write the Java Software. It is also can support different types of multiple languages and other features into any default packages as well as provides unlimited extension and customization.

iv. Gradle

Gradle is a build automation tolls that providing flexibility to build a software and also automating the creatin of applications. It is very useful to implement in languages such as Java, Android, C/C++, Scala, etc. Since it is highly customized, hence it provides a better use experience and easier managed a wide variety of IDE. This tool mainly handles on usability, extendibility, maintainability and performance.

v. Sphinx4

Sphinx4 is the core-framework that provides modular, flexible, and pluggable structure to support new innovations in the core research of Hidden Markov model (HMM) of the speech recognition. It is also open sources tools for researchers with a “research ready system” to exercise their groundwork and involving different kinds of implementations for the techniques based on its designs and patterns. It is also adjustable and modifiable recognizer that written in Java model.

vi. Notepad++

It is an open-source text editor which can featuring syntax, auto complete, code folding and highlighting and especially auto completion for markup, scripting, and programming languages. It also can be hosted on GitHub to distribute the projects with multiple working windows.

vii. Visual Studio Code

XAMPP Visual Studio code is a code editor optimized and redefined for constructing and debugging the modern cloud and web applications. It is free and built on the open source, and can even run everywhere, anytime. Its extensions can be implemented in separate processes.

viii. Anaconda Navigator/ anaconda3

Anaconda navigator is a GUI tool which consists of pre-packaged distribution of Anaconda such as Python. It supports setting up the applications and handles conda packages, channels and environments without applying command-line commands easily. Due to it provides multiple versions of various packages and can deploy at multiple environments to detach these different versions, the application of Jupyter notebook is recommended to install.

ix. Jupyter notebook

Jupyter notebook is an open-source web application that can be deployed for different kinds of use cases and user stories. For the example in use are numerical simulation, statistical modelling, data visualization, data cleaning and transformation, machine learning and so on. It also allows you to do several tasks like create and share documents that include equations, visualizations, live code and narrative text. This tool is convenient yet useful for classification tasks based on sentiment analysis.

x. Markup language

Development of an interactive yet user-friendly website is deployed by using the HTML and CSS scripts. HTML will be built for the text-based content on the website and created for aligning and locating the text and output in the most efficient way. While the CSS is performed to make the website stylish, attractive and user friendly to the users.

CHAPTER 5: SYSTEM IMPLEMENTATION

For this project, the Python programming language is implemented to construct the web application's backend, while the flask is deployed to be the web framework. The libraries and packages needed to create the system are as follows:

i. Flask

Flask is a lightweight web framework in Python that generates effective features and tools that can build web applications in Python in an easier and shorter time. Its flexibility supports the developers for accessing the framework to the new learners by implementing a single input Python file only.

ii. Flask-RESTful

Flask-RESTful is an extension for Flask that supports the extension to construct the REST APIs deployment in Python. It is good for supporting writing clean object-oriented code and very easy to set up for building blocks.

iii. Flask-CORS

Flask-CORS is known as the Cross Origin Resource Sharing of the flask extension that creates the cross-origin AJAX possible. It is also determined by the circumstances while the domain requesting a resource that is not the same from the domain serving that resource.

iv. NLTK (Natural Language Toolkit)

NLTK is a leading platform for developing Python programs to deploy the human language data. NLTK is a complimentary and open source yet community-driven program by bringing some useful interfaces for corpora and lexical resources. It also provides a series of packages and programs based on Natural language processing (NLP) to handle some text processing tasks like tokenization, classification, tagging etc.

v. Scikit-learn framework

Scikit-learn is one of the machine learning tools for solving classification issues in python. It is a high-level framework to employ off-the-shelf machine learning algorithms rather than create a new one. Due to its lack of customizability, it is easy for training classifiers in just a few lines of codes immediately.

vi. Gensim

Gensim is a python library that can train large scale semantic NLP models. It is also an NLP package which is quite suitable for LDA topic modeling and other work embedding machine learning algorithms. It also represents text as semantic vectors and searches related documents and content semantically.

vii. Spacy

Spacy is one of the open source software in the python library that is implemented in advanced natural language processing and machine learning. It is always carried to construct the information extraction, preprocessing text for deep learning and natural language understanding system.

viii. WordCloud

The WordCloud is the visualization for displaying the text data, which is also known as text clouds or tag clouds). In the simpler way it is text analysis to display the specific words that occur in a source of textual data such as database, speech, blog post, article etc frequently and notoriously.

There are some libraries needed to install at Anaconda Navigator or at the Anaconda Command Prompt. Here are the following commands and details in the table:

Library	Command	Remark
SpeechRecognition	pip install Speech Recognition	Speech to text recognition
Flask	pip install Flask	Web application framework
Flask-CORS	pip install flask-cors	Flask extension for CORS
Flask-RESTful	pip install flask-restful	Flask extension for RESTful APIs
Gensim	pip install gensim	Gensim library
Spacy	pip install spacy	Spacy library for en_core_web_sm
Wordcloud	pip install wordcloud	Wordcloud library
nlTK	pip install nltk	Wordcloud nltk

Table 5.2: Installation Python libraries

5.3 Setting and Configuration

5.3.1 Flask Configuration

```

app = Flask(__name__)

CORS(app)
api = Api(app)
class determineToxic(Resource):
    def post(self):
        if request.method == 'POST':
            f = request.files['file']
            f.save(secure_filename(f.filename))
            print(f)
            print(f.filename)
            text,text_arr = get_large_audio_transcription(f.filename)
            print(text)
            write_file(text)
            result = getResult()
            response = json.dumps({"non-toxic": result[0][0], "toxic": result[0][1], "text": text})
            return response

api.add_resource(determineToxic, '/determineToxic')

if __name__ == '__main__':
    app.run(debug = False)

* Serving Flask app "__main__" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off

* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

```

Figure 5.3.: Flask framework application URL

The figure above has demonstrated that the last line `http://127.0.0.1:5000/` is the one of default URL and port number in the application as the IP address with the port number of 5000. This also determines that the web server is running on port 5000 and accessible to the browser. We can modify the code according the requirements is needed to add on the settings since no additional setting on the respective hosting address and port number. For the instance, with the following of the the address is implied at thost parameters while the port number implied in the port address, the `app.run` in `host = "128.0.0.0"` with the `port = 8888` and it will indicate to the address of <http://128.0.0.0:8888/> as an example.

5.3.2 API Configurations

```
app = Flask(__name__)  
  
CORS(app)  
api = Api(app)
```

Figure 5.4: API Configuration

With figure following, the API configuration has been carried out with `CORS(app)` to support this library on every route. Hence, every CORS header is set up on the per-resource level to the parameterize and then it will authorize this method CORS for all domains on the every route in brief.

5.4 Implementation of System

According to the proposed system design for the speech to text, this project has selected Google Speech Recognition by using python to transcript the audio file to the text file. Also, the text classification models will be carried out as mentioned in Chapter 4 for further training and testing the speech-to-text of an individual. In order to let users interact with the Toxic Friend detector system, a web application is designed to present the product of speech to text and text classification for the toxic and non-toxic labeled on the webpage. The Flask framework for the REStful API and the other packages

library is implemented in order to connect and communicate between the frontend and the backend side.

5.5 System Operation

5.5.1 Speech-To-Text Operation

```
# importing libraries
import tensorflow as tf
import numpy as np
import speech_recognition as sr
import os
from pydub import AudioSegment
from pydub.silence import split_on_silence
```

Figure 5.5: Speech Recognition libraries

```
# create a speech recognition object
r = sr.Recognizer()
```

Figure 5.6: Speech Recognition libraries

The figure above has shown that the importing speech_recognition libraries and packages since every instance will be deployed with a sort of settings and functionality for recognizing the speech from an audio file source.

```
def get_large_audio_transcription(path):
    """
    Splitting the large audio file into chunks
    and apply speech recognition on each of these chunks
    """
    text_arr = []
    # open the audio file using pydub
    sound = AudioSegment.from_wav(path)
    # split audio sound where silence is 700 milliseconds or more and get chunks
    chunks = split_on_silence(sound,
        # experiment with this value for your target audio file
        min_silence_len = 500,
        # adjust this per requirement
        silence_thresh = sound.dBFS-14,
        # keep the silence for 1 second, adjustable as well
        keep_silence=500,
    )
```

Figure 5.7: Code snippet of Transcription speech to text

```
)
folder_name = "audio-chunks"
# create a directory to store the audio chunks
if not os.path.isdir(folder_name):
    os.mkdir(folder_name)
whole_text = ""
# process each chunk
for i, audio_chunk in enumerate(chunks, start=0):
    # export audio chunk and save it in
    # the `folder_name` directory.
    chunk_filename = os.path.join(folder_name, f"chunk{i+1}.wav")
    audio_chunk.export(chunk_filename, format="wav")
    # recognize the chunk
    with sr.AudioFile(chunk_filename) as source:
        audio_listened = r.record(source)
        # try converting it to text
        try:
            text = r.recognize_google(audio_listened)
        except sr.UnknownValueError as e:
            print("Error:", str(e))

        else:
            text = f"{text.capitalize()}. "
            print(chunk_filename, ":", text)
            text_arr.append(text)
            whole_text += text
# return the text for all chunks detected
return whole_text, text_arr
return whole_text
```

Figure 5.8: Code snippet of Transcription speech to text

The diagram above demonstrates the code snippet of the function of retrieving the large audio file and then transcript the speech to text with one of the audio source APIs which is recognize_google() with the Google Web Speech API.

```

path = "kw_full.wav"
text,text_arr = get_large_audio_transcription(path)
print("\nFull text:", text)

audio-chunks\chunk1.wav : Is ok i know it's my fault.
audio-chunks\chunk2.wav : Hey pretty girls i know all this.
audio-chunks\chunk3.wav : Hey girls i'm not trying to argue with you and i don't want to fight in this war.
audio-chunks\chunk4.wav : Singing.
audio-chunks\chunk5.wav : Are you busy on this friday.
audio-chunks\chunk6.wav : Is it ok if i request to stay overnight.
audio-chunks\chunk7.wav : Of course you can eat.
audio-chunks\chunk8.wav : Thank you so much.
audio-chunks\chunk9.wav : I have a physical discussion of assignment on friday.
audio-chunks\chunk10.wav : I'm not sure i will discuss aunty what time.
audio-chunks\chunk11.wav : I will make nice also no worries.
audio-chunks\chunk12.wav : No.
audio-chunks\chunk13.wav : Asleep already marsh appreciate.
audio-chunks\chunk14.wav : I will cellular call you.
audio-chunks\chunk15.wav : Is this.
audio-chunks\chunk16.wav : Gum q1.
audio-chunks\chunk17.wav : Do you still need.
audio-chunks\chunk18.wav : Are you following for dinner.
audio-chunks\chunk19.wav : I mean my car.
audio-chunks\chunk20.wav : I will make nice also no worries.

print(text_arr[10])

I will make nice also no worries.

```

Figure 5.9: Result of Transcription speech to text

The figure illustrates that the function proceeds with the Audio files. However, this Speech Recognition can only support a few audio file formats such as .wav(must be in PCM/LPCM format), AIFF, AIFF-C and FLAC (which is under native FLAC format or else the OGG-FLAC is not supported). It also successfully transcribes the audio file one by one by displaying it in text form.

```

#import to excel spreadsheet.xlsx

import xlswriter

workbook = xlswriter.Workbook('kw_full_1.xlsx')
worksheet = workbook.add_worksheet()

```

Figure 5.10: Export as Text Excel format

A	B	C	D	E	F	G	H
id	comment_text						
1	Is ok i know it's my fault.						
2	Hey pretty girls i know all this.						
3	Hey girls i'm not trying to argue with you and i don't want to fight in this war.						
4	Singing.						
5	Are you busy on this friday.						
6	Is it ok if i request to stay overnight.						
7	Of course you can eat.						
8	Thank you so much.						
9	I have a physical discussion of assignment on friday.						
10	I'm not sure i would discuss auntie work time.						
11	I will make nice also no worries.						
12	No.						

Figure 5.11: Transcription text in excel

The following two figures above demonstrate the text result into the excel spreadsheet under .xlsx format for further deploying as texting data in text classification model.

5.5.2 Text Classification Operation

Importing Tools

```
# UTILITY
from PIL import Image
from joblib import dump
import pandas as pd
import numpy as np
from collections import Counter
from sklearn.preprocessing import MinMaxScaler, Normalizer
import warnings

# IMBALANCED LEARN
from imblearn.over_sampling import SMOTE

# NLP
import re
from nltk.corpus import wordnet
from nltk.tokenize import word_tokenize
from nltk import SnowballStemmer
from nltk import punkt
import spacy
from gensim.models.doc2vec import Doc2Vec, TaggedDocument, Word2Vec
from gensim.test.utils import get_tmpfile
from sklearn.feature_extraction.text import TfidfVectorizer

# MODELS
from sklearn.model_selection import train_test_split, RandomizedSearchCV
from sklearn.svm import LinearSVC
```

Figure 5.12: Installation of libraries and packages

Figure has shown several tools and libraries being downloaded as under the requirements for implementing this project.

Importing Data and Cleaning

```
def apply_regex(corpus):
    corpus = corpus.apply(lambda x: re.sub("\S*\d\S*", " ", x)) # removes numbers and words concatenated with numbers
    corpus = corpus.apply(lambda x: re.sub("\S*@\S*\s?", " ", x)) # removes emails and mentions (words with @)
    corpus = corpus.apply(lambda x: re.sub("\S*#\S*\s?", " ", x)) # removes hashtags (words with #)
    corpus = corpus.apply(lambda x: re.sub(r'http\S+', ' ', x)) # removes URLs
    corpus = corpus.apply(lambda x: re.sub(r'[^\w-zA-Z0-9_]', ' ', x)) # keeps numbers and letters
    corpus = corpus.apply(lambda x: x.replace(u'\ufffd', '8')) # replaces the ASCII '�' symbol with '8'
    corpus = corpus.apply(lambda x: re.sub(' +', ' ', x)) # removes multiple spaces
    return corpus
```

Figure 5.13: Data Cleaning

Figure above has performed the basic regex-based text-cleaning operations.

Preprocessing Text - NLP Pipeline

data.sample(3)

	id	comment_text	toxic	label	tokenized	stemmed
140863	f1c9f4149bb40ffc	I think autistics like you can be an asset in ...	0	0	[think, autistics, like, asset, situations, em...	[think, autist, like, asset, situat, emot, clo...
80489	d759981f9eec2cad	REDIRECT Talk Liberty Charter High School La M...	0	0	[redirect, talk, liberty, charter, high, schoo...	[redirect, talk, liberti, charter, high, schoo...
16218	2ac37bd44d83bac2	Early Life Can someone confirm citation for e...	0	0	[early, life, can, confirm, citation, early, l...	[earli, life, can, confirm, citat, earli, life...

Figure 5.14: Result of NLP processing

Figure above shows the result of all the steps from the raw text to the stemmed text. It proceeds with the NLP Pipeline by tokenization, stop word removal and stemming the tokens of those techniques.

Feature Visualizations

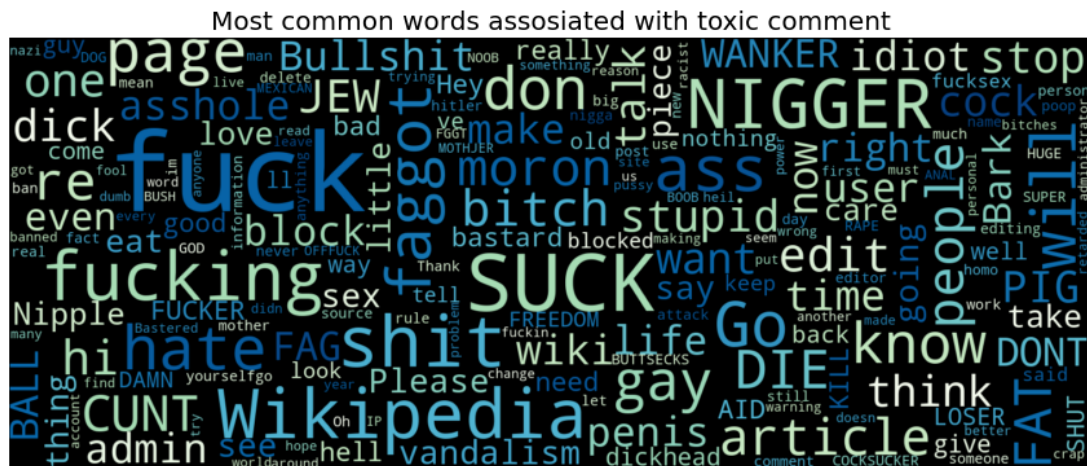


Figure 5.15: Word Clouds of the toxicity of the words

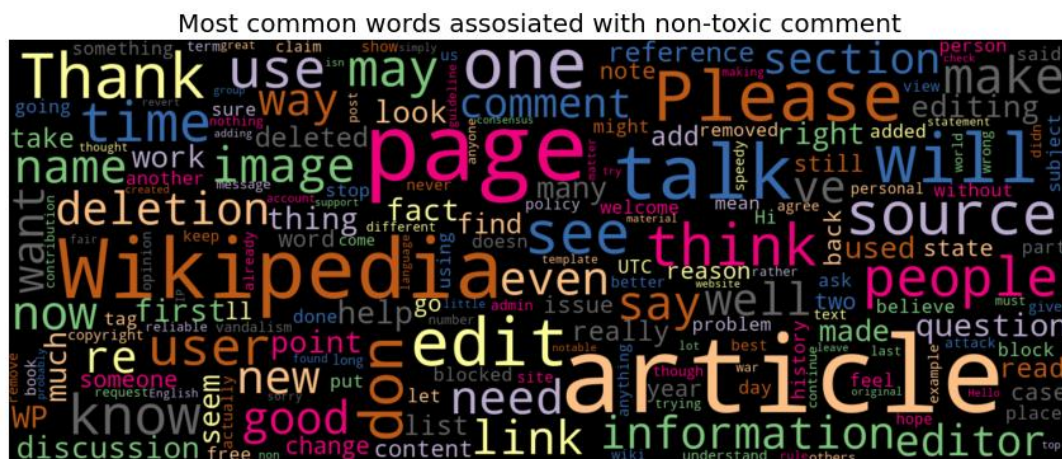


Figure 5.16: Word Clouds of the non-toxicity of the words

Both figures above show the Wordcloud of the toxic words and the non-toxic that commonly being generated in its library during the text analysis. Those words that are frequent and prominent in the body of the text will be deployed as following figures.

Splitting the Data in Training and Testing sets

```
data["stemmed"]
0      [explan, whi, edit, usernam, hardcor, metallic...
1      [aww, he, match, background, colour, seem, stu...
2      [hey, man, tri, edit, war, it, guy, constant, ...
3      [more, real, suggest, improv, wonder, section,...
4      [you, sir, hero, ani, chanc, rememb, page]
...
159566 [and, second, time, ask, view, complet, contra...
159567 [you, asham, that, horribl, thing, talk, page]
159568 [spitzer, umm, there, actual, articl, prostitu...
159569 [and, look, like, actual, speedi, version, del...
159570 [and, don, think, understand, came, idea, bad,...
Name: stemmed, Length: 159571, dtype: object
```

Figure 5.17: Output of the stemmed data

Figure above shows the stemmed data will be used for splitting the dataset into train and test sets with the function of `train_test_split` by sklearn.

Feature Engineering

```
# encodin text into vectors
tfidf = TfidfVectorizer(lowercase=False, tokenizer=do_nothing, max_features=500)

train_vectors_tfidf = tfidf.fit_transform(X_train).toarray()
test_vectors_tfidf = tfidf.transform(X_test).toarray()
```

Figure 5.18: Vectorization of TF-IDF

```
# building the vocabulary and training the model
w2v.build_vocab(X_train, progress_per=50000)
w2v.train(X_train, total_examples=w2v.corpus_count, epochs=30, report_delay=1)

# saving the model to the disk in order to avoid training again
w2v.save("w2vec.model")
print("Model Saved")

Model Saved
```

Figure 5.19: Vectorization of Word2Vec

```
# building the vocabulary and training the model
model.build_vocab(train_doc2vec)
model.train(train_doc2vec, total_examples=model.corpus_count, epochs=model.epochs)

# saving the model to the disk in order to avoid training again
model.save("d2v_comments.model")
print("Model Saved")

Model Saved
```

Figure 5.20: Vectorization of Doc2Vec

With the following 3 figures above, the three different embeddings TF-IDF, Word2Vec and Doc2Vec have saved their model with different methods for encoding the stemmed text.

Handling with Imbalanced Class

```
# SMOTE for the tf-idf encoding
sm_tfidf = SMOTE(random_state=42, n_jobs=NUM_OF_THREADS)
train_vectors_tfidf, y_train_tfidf = sm_tfidf.fit_resample(train_vectors_tfidf, y_train)
test_vectors_tfidf, y_test_tfidf = sm_tfidf.fit_resample(test_vectors_tfidf, y_test)

# SMOTE for the w2v encoding
sm_w2v = SMOTE(random_state=42, n_jobs=NUM_OF_THREADS)
train_vectors_w2v, y_train_w2v = sm_w2v.fit_resample(train_vectors_w2v, y_train)
test_vectors_w2v, y_test_w2v = sm_w2v.fit_resample(test_vectors_w2v, y_test)

# SMOTE for the D2V encoding
sm_d2v = SMOTE(random_state=42, n_jobs=NUM_OF_THREADS)
train_vectors_d2v, y_train_d2v = sm_d2v.fit_resample(train_vectors_d2v, y_train)
test_vectors_d2v, y_test_d2v = sm_d2v.fit_resample(test_vectors_d2v, y_test)
```

Figure 5.21: Different embeddings with SMOTE

Figure has shown several embeddings with the SMOTE (synthetic minority oversampling technique) algorithm to generate new synthetic data points in order to approach the imbalance of the dataset.

Feature normalization

```
# NORMALIZING DOC2VEC VECTORS
norm_D2V = Normalizer(copy=False)
norm_train_d2v = norm_D2V.fit_transform(train_vectors_d2v)
norm_test_d2v = norm_D2V.transform(test_vectors_d2v)
```

```
test_vectors_d2v
```

```
[[0.09604042768478394,
 -0.08733847737312317,
 -0.21641018986701965,
 -0.30469512939453125,
 0.44060802459716797,
 -0.10342638939619064,
 -0.3441833257675171,
 0.1853000819683075,
 -0.24785128235816956,
 0.06476390361785889,
 -0.15825535356998444,
 0.2614579498767853,
 0.07588884234428406,
```

Figure 5.22: Normalization of DocVec

Figure has demonstrated one of the embeddings, Doc2Vec has normalized the data for classfi in further text classification.

Algorithm Classifier Selection

```
models.append(('LinearSVC', LinearSVC(random_state=seed, class_weight=classW)))
models.append(('Logit', LogisticRegression(random_state=seed, class_weight=classW, n_jobs=NUM_OF_THREADS, max_iter=300)))
models.append(('NaiveBayesMN', MultinomialNB()))
```

Figure 5.23: Algorithm classifiers models

Figure has performed a custom function to test programmatically with three different baseline models which is Linear SVC baseline, Logistic Baseline and Naive Bayes baseline with each encoding methods used(TF-IDF, Word2Vec and Doc2Vec)

5.5.3 Web Application Operation

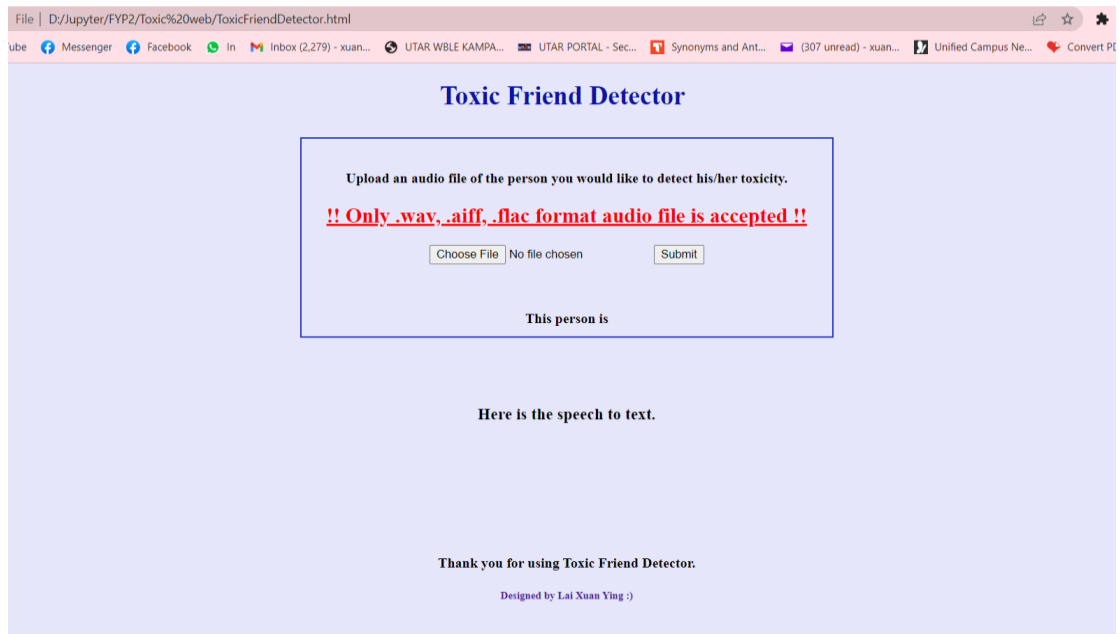


Figure 5.24: Web Page of System's Web Application

The figure above has shown a simple GUI design for users to interact with the Toxic Friend Detector system. There are some messages that are described to let the users read and use the system.

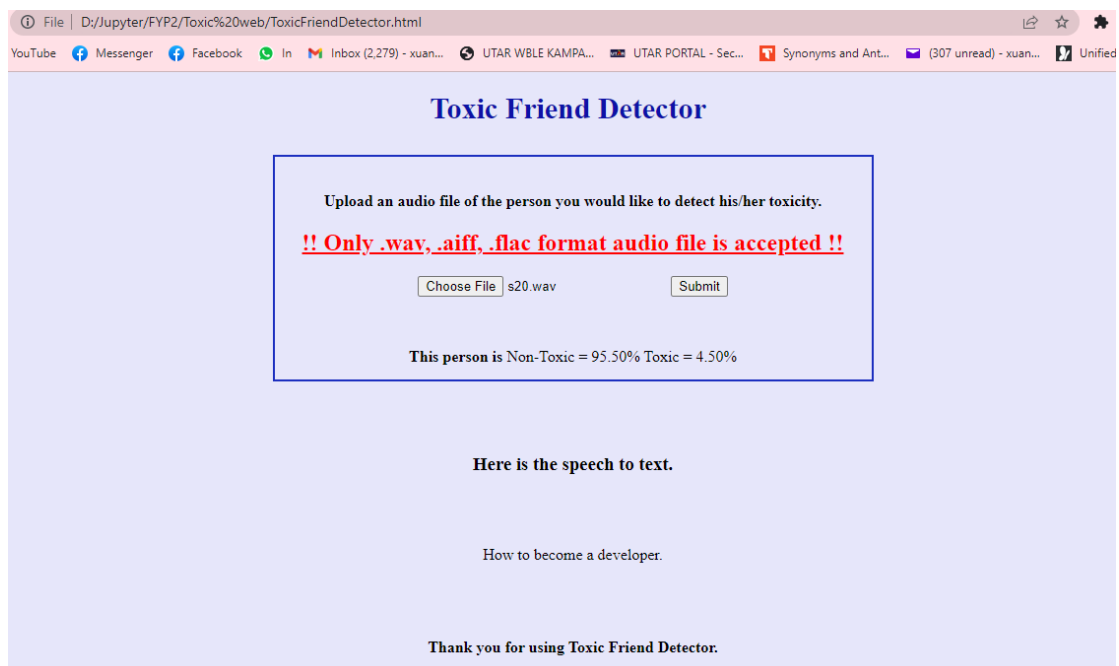


Figure 5.25: Display Output

CHAPTER 5: SYSTEM IMPLEMENTATION

The figure above has shown the output which is Non-Toxic = 95.50% and Toxic = 4.50% and the speech to text output after the user has submitted the input speech file.

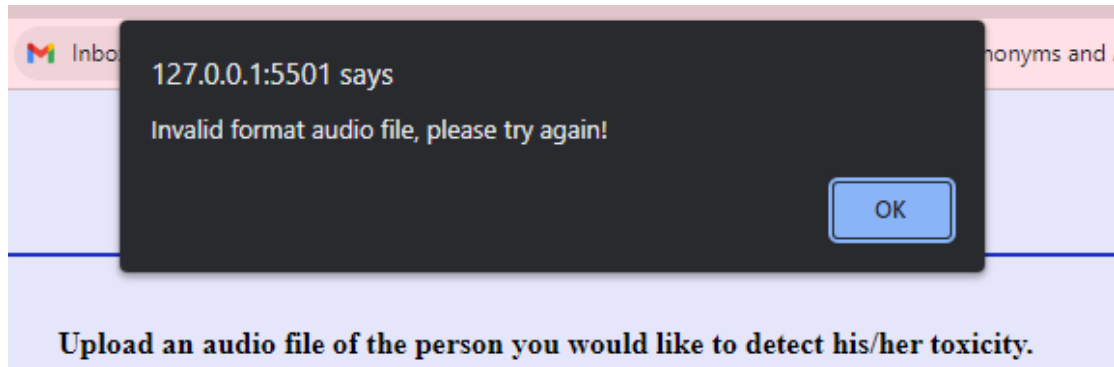


Figure 5.26: Display alert message

The figure above shows an alert message if the user has uploaded invalid format audio file.

CHAPTER 6: SYSTEM EVALUATION AND DISCUSSION

6.1 System Testing and Performance Metrics

In order to test the TF-IDF, Word2Vec and Doc2Vec embeddings with the different baseline models including the Linear SVC baseline, Logistic Regression baseline and Naives Bayes baseline, there will be 9 model combinations for further fine-tuning. Several evaluation metrics such as Accuracy, F1 score, Recall, Precision Specificaticity and so on to evaluate the performance of different techniques and models.

The following figures show the results of 9 models approaching each other.

```
results = test_models(models, norm_train_tfidf, y_train_tfidf, norm_test_tfidf, y_test_tfidf)
results
```

	Model	Accuracy	f1	Recall	Precision	Specificity	TP	TN	FP	FN	y_test size
0	LinearSVC	0.835260	0.827422	0.789844	0.868754	0.880675	28496	31773	4305	7582	72156
0	Logit	0.836050	0.828759	0.793475	0.867327	0.878624	28627	31699	4379	7451	72156
0	NaiveBayesMN	0.829605	0.820178	0.777177	0.868215	0.882033	28039	31822	4256	8039	72156

Figure 6.1: Performance measurement of every algorithm classifier with TF-IDF

Figure above has shown the dataframe with classification metrics and confusion matrix values after testing with embedding TF-IDF approach.

```
results = test_models(models, norm_train_w2v, y_train_w2v, norm_test_w2v, y_test_w2v)
results
```

	Model	Accuracy	f1	Recall	Precision	Specificity	TP	TN	FP	FN	y_test size
0	LinearSVC	0.928599	0.927892	0.918787	0.937178	0.938411	33148	33856	2222	2930	72156
0	Logit	0.928100	0.927343	0.917678	0.937213	0.938522	33108	33860	2218	2970	72156
0	NaiveBayesMN	0.862284	0.858004	0.832141	0.885526	0.892428	30022	32197	3881	6056	72156

Figure 6.2: Performance measurement of every algorithm classifier with Word2Vec

Figure above has shown the dataframe with classification metrics and conviction matrix values after testing with embedding Word2Vec approach. .

```
results = test_models(models, norm_train_d2v, y_train_d2v, norm_test_d2v, y_test_d2v)
results
```

	Model	Accuracy	f1	Recall	Precision	Specificity	TP	TN	FP	FN	y_test size
0	LinearSVC	0.929639	0.928817	0.918094	0.939793	0.941183	33123	33956	2122	2955	72156
0	Logit	0.929902	0.929086	0.918399	0.940025	0.941405	33134	33964	2114	2944	72156
0	NaiveBayesMN	0.897375	0.892655	0.853401	0.935694	0.941349	30789	33962	2116	5289	72156

Figure 6.3: Performance measurement of every algorithm classifier with Doc2Vec

Figure above has shown the dataframe with classification metrics and confusion matrix values after testing with embedding Doc2Vec approach.

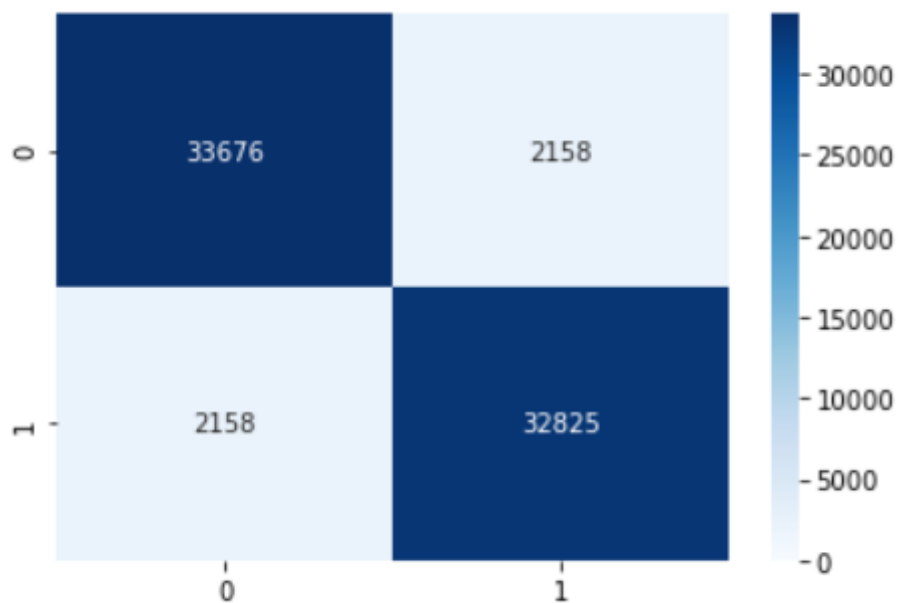


Figure 6.4: Confusion Metric of logistic regression with the Doc2Vec

Since the logistic regression with the Doc2Vec has a better performance for accuracy, the confusion matrix of this embedding technique is displayed. The results shows that True positive (TP) has 32825, while the True Negative(TN) has 33676, Also, the False Negative (FN) and False Positive (FP) has same value of the 2158

6.2 Testing Setup and Result

6.2.1 Randomized Search to search the best parameters

After having the result of the overall evaluation performance of the 9 approach models, Logistic Regression with Doc2Vec is identified as better encoding and better model amongst the embedding approach. The hyperparameter search will be deployed for searching the best parameters. For instance, we will construct a parameter grid for the model and run the randomized search with the cross validation folds established to the 8 and 15 iterations.

The following figures show the results of 9 models approaching each other.

```
# return the possibly best estimator
final_model.best_estimator_

LogisticRegression(C=5, max_iter=800, random_state=42, solver='newton-cg',
                    tol=0.04)
```

Figure 6.5: Result of the possibly best estimator

Figure above has shown the result of the possibly best estimator after running the search and fitting the model and making predictions with the testing data in order to compute the score.

6.2.2 Performance of the model proposed

To evaluate the performance of the Logistic Regression with the Doc2Vec approach, the probability of related to a class is converted into the finalized binary class with the particular threshold. Also, through the dataframe with all the meters for the evaluation of the model. A custom function such as the probability from `predict_proba()` and a threshold value will be deployed to convert the probability to a binary class [0,1]. Assumption that probability is part of the “1” class which indicates it is toxic text by using the threshold t . The model has set the threshold as $t=0$. for consistent result as it is default threshold that is deployed by the algorithm.


```
predictions=probs_to_prediction(probability, 0.5)
make_classification_score(y_test_d2v, predictions, "logit, t=0.5")
```

	Model	Accuracy	f1	Recall	Precision	Specificity	TP	TN	FP	FN	y_test size
0	logit, t=0.5	0.886246	0.877221	0.812739	0.952817	0.959754	29322	34626	1452	6756	72156

Figure 6.6: Result Score of Logistic Regression with Doc2Vec

The result is not bad since all considered metrics are over .81. However, there are still several False Positives and False Negatives that influence a better threshold tuning to maximize the metric.

6.2.3 Result on Performance Measurement

6.2.3.1 ROC AUC and Curve Plot

The Area Under The roc Curve (AUC) has been computed to identify whether it is a perfect classifier or random classifier. By definition that AUC = 1 indicates to a perfect classifier while the AUC = 0.5 indicated to a random classifier.

```
# calculate roc auc score
print("SVM: ROC AUC = %.4f" % roc_auc_score(y_test_d2v, probability[:, 1]))

SVM: ROC AUC = 0.9608
```

Figure 6.7: Result of AUC

The figure has shown that our AUC value is 0.9608 which is nearer the AUC = 1, thus, it is not a bad classifier.

ROC curve is known as the receiver operating characteristics curve act as a graph indicating the performance of a classification model at the most classification thresholds.

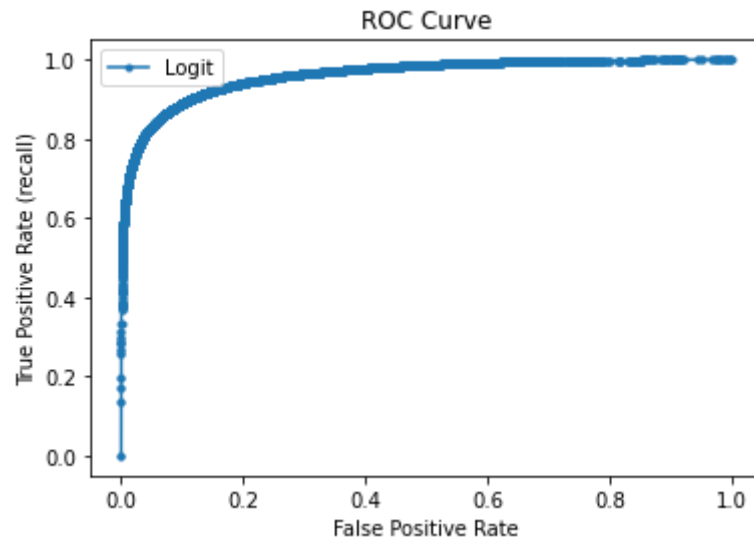


Figure 6.8 : Result of ROC Curve

Diagram is illustrated with the ROC curve whereas the dots are indicated to the respective different threshold values.

6.2.3.2 Precision-Recall Curve Plot

According to the unbalanced classes, a Precision-Recall curve is deployed rather than ROC curve. Precision-recall curve has demonstrated the tradeoff between precision and recall for an unsimilar threshold. The value of the threshold can be picked up matching to the desired values of the precision and recall.

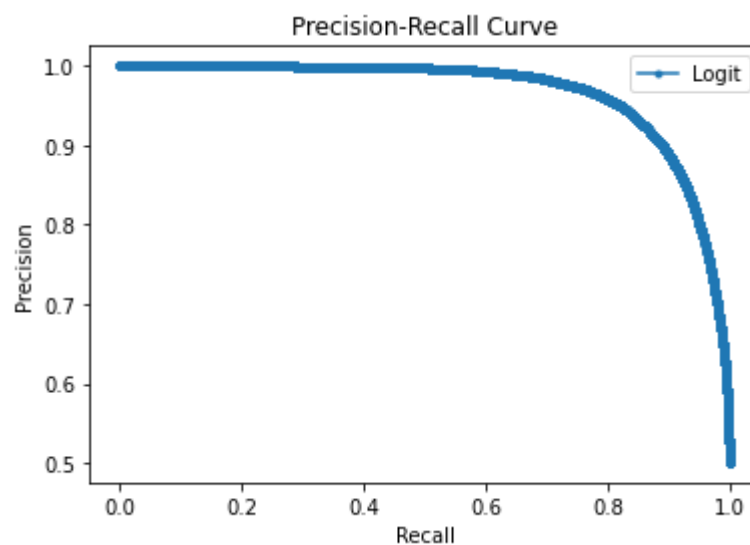


Figure 6.9 : Precision-Recall Curve

6.2.3.2 Threshold Tuning

To expand the precision versus the recall for the metric according to the domain and the application itself. Depending on this project that we have implemented for toxic text classification model, this it is better to maximize according to the cost of a false positive or false negative which means there is no a right or wrong answer there.

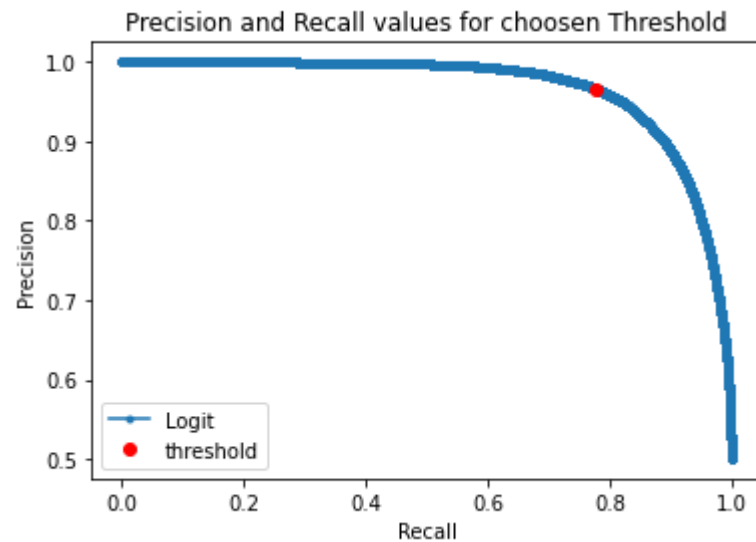


Figure 6.10 : Precision and Recall values for chosen Threshold

The diagram has shown that the number of false positives with the proposed threshold value is 0.7495 and the precision is 0.9793. On the contrary the recall result has dropped to 0.7414.

CHAPTER 6: SYSTEM EVALUATION AND DISCUSSION

6.2.4 Web application testing



Figure 6.11: Display output at Web page

The diagram has demonstrated the web application testing with correct input speech file. The output has displayed the toxicity from the speech file and text transcription of the speech.

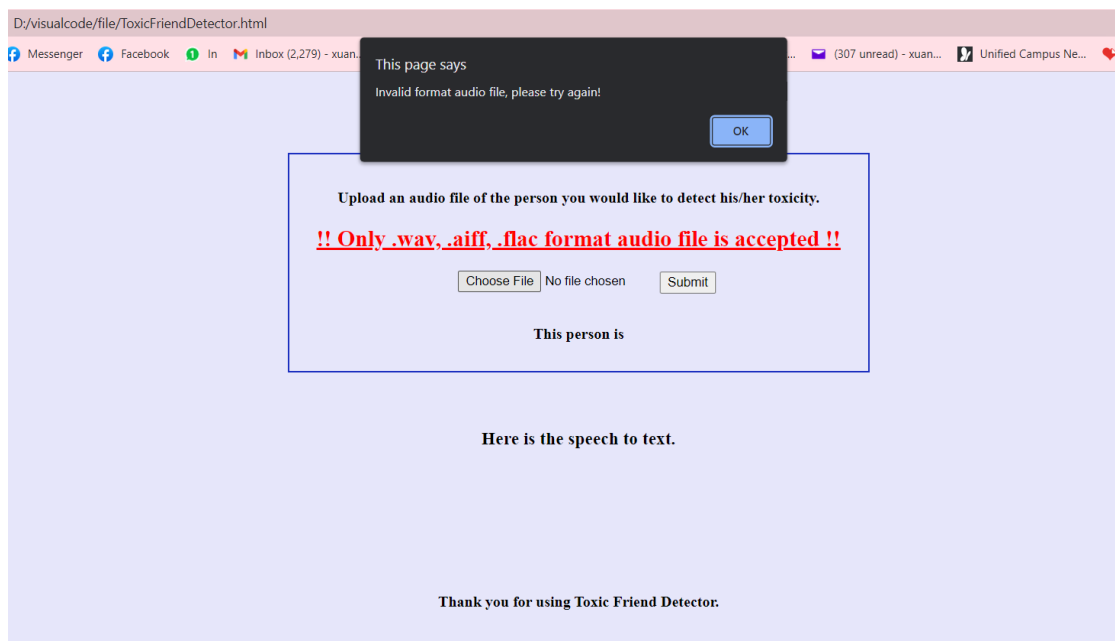


Figure 6.12 : Display alert message at Web page

CHAPTER 6: SYSTEM EVALUATION AND DISCUSSION

The diagram has demonstrated the web application with incorrect input file format. The error message will be displayed to alert the users that wrong input to recognition.

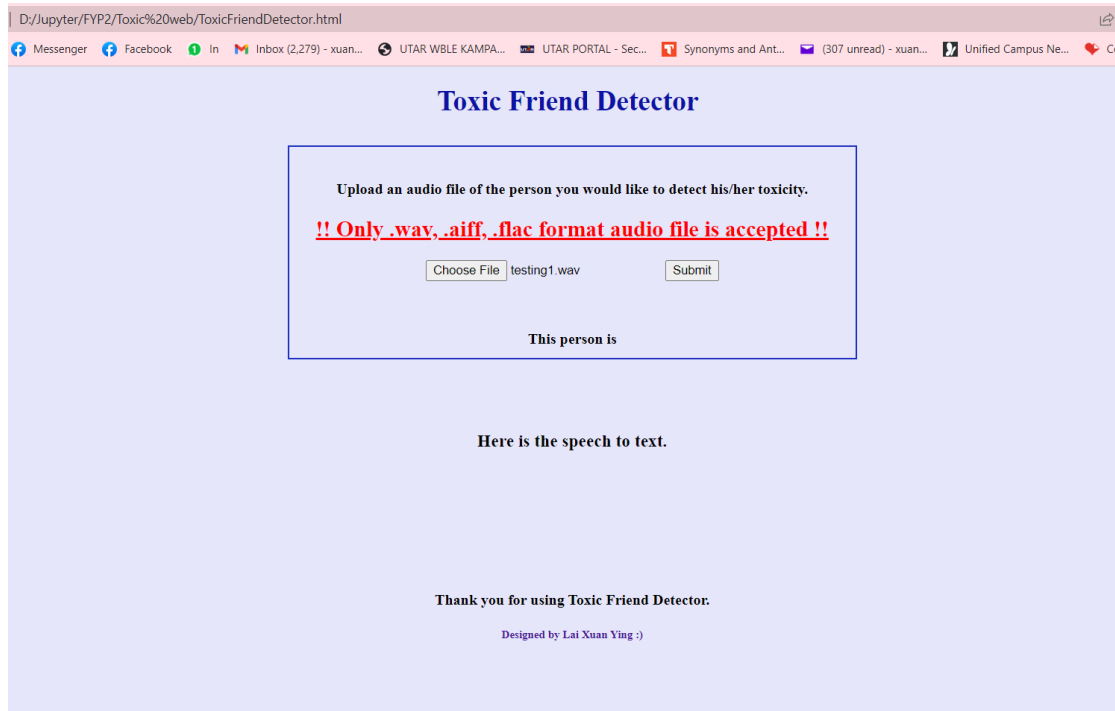


Figure 6.2.4.3 : Nothing change after submitting the input file

Also, a large speech file will cause the API response invalid and the number will be bigger which means it failed to connect with the backend model.

6.3 Project Challenges

There are several challenges that have been met and some difficulty in constructing this project.

- a. Speech files are collected by recording from the real-time environment between the individuals. The recording may include some noise and effects from the environment.
- b. Due to unfamiliarity about the API, lots of the practices need to take to understand the API connection with the machine learning model. There are many times of trying and practicing to further understand it.
- c. The speech file format is limited to a few specific formats only. There is the limitation when during
- d. The system may not perform well, for example, when the user uploads the large speech file, the model requires plenty of time to run and display the output.
- e. Also, the model accuracy of the proposed speech to text model of the Sphinx4 is quite low among the existing speech recognition. Thus, the model

6.4 Objectives Evaluation

The proposed project has reached the goal for the project objectives:

- I. The speech and text data have been performed well in the filtering and resolving of the natural language processing to execute and filter the unnecessary and noise data. The data also has been successful to be resolved with some techniques such as tokenization and POS for data cleaning and unification.
- II. The data can be successfully trained and tested by deploying the machine learning methods based on sentiment analysis. The Logistic regression embedding with the Doc2Vec will be carried out to classify the content for feature extraction, normalization and classification for the classes on the toxic and non-toxic labels.
- III. A detecting personality system in the form of a web application with a simple user interface (UI) is well built with connecting to the back end models. The application has a design with two alignments for input and output which is user-friendly. The input can be successfully inserted and the outcome of the result is successfully displayed whether this individual is toxic or not.

6.5 Concluding Remark

Throughout the main functions are the speech to text and classification of toxicity of an individual are still not in a great recognition and classification model at predicting the speech and class of the frame. There are several possible reasons that result in poor performance.

- i. There might be inadequate and insufficient data for the model to learn and construct well
- ii. There might be some bad hyperparameters settings being chosen to train the proposed model
- iii. There might be occurs the inconsistency or problems in labeling the dataset
- iv. There might be some of the techniques or methods not work well in the system.

CHAPTER 7: CONCLUSION AND RECOMMENDATION

7.1 Conclusion on Project Achievements

In this increasingly social media-driven world, social networks have been a widely used and trendy medium for facilitator of social interactions and information dissemination. Due to the emergence of online social networks, people nowadays often use online social media dramatically in a way of expressing their feelings, opinions, characteristics, behaviors and preferences freely without being judged. It is not easy to understand or well know an individual's personality and behavior over time and situation. Typically, this project will be developed to explore and predict personality traits based on these data contents from users' text and speech conversation.

The motivation of this project is to recognize the speech from the text according to the communication between individuals by deploying a speech recognizing system. These outcomes can support to spot different kinds of people and determine who is a toxic person from different perspectives. Hence, the system under calculation and automation-based algorithms modeling and several various related technologies methods to construct the sentiment analysis. And the personality traits detection. It is good to find out toxic people and escape to these toxic people for enhancing personal development and encouraging a sense of belonging from our own. [17]

This paper studies automatic speech recognition by using the English Language environment. The recognition and detection of speech has been divided into several processes such as data collection, data preprocessing, data post processing, decoding and result extraction. During our experiment, it can demonstrate the CMU Sphinx4 adaptability for English Language with the large vocabulary during the post-processing process. While the use of a phonetic dictionary will contain all phonetic the words sound for conducting in training, while the use of language model will provide the probability of the sequence of the words. Also, the acoustic model will conduct the HMM that serving as each phoneme in unit. However, in order to have a better performance and accuracy of the Speech recognition system, the Google Speech recognition is carried out to replace the proposed Sphinx4 Speech recognition.

Until this stage, the new proposed google speech recognition model is well generalized and accomplishes a good result for the performance in terms of flexibility and accuracy. The text classification will continue to be developed and constructed in

CHAPTER 7: CONCLUSION AND RECOMMENDATION

the training and recognition phase. The tokenization, stopwords and stemming is carried out in the preprocessing and the TF-IDF, Word2Vec, Doc2Vec will also implemented in feature engineering for embedding with the several algorithm classifiers such as Linear SVC, Naives Bayes and Logistic Regression. In this project, the Logistic Regression with the Doc2Vec is chosen to perform the text classification and as the final deployment with the most optimal hyperparameter setting with the low validation loss. The accuracy of 88.82% which is well-performed in other models. In contrast, it is critical to emphasize the need of optimization to make the model to be more efficient to perform the same task. Although the project was completed at the point of writing, the current work denoted the fairly good progress, and quite confidentially to complete and achieve the project goal.

7.2 Recommendation

The project can be further improved to support different types or sorts of toxicity of an individual with a larger dataset to achieve a more practical model for real life usage. However, the possible recommendation can be given is to implement better algorithm classifiers such as CNN and the LSTM to feed into the text classification model. Besides, for the speech recognition model, other languages can be included in the future work or other speech recognition with better performance can be employed to get better results. Also, the web application part may be developed with different kinds of features such as emoji etc, to be attractive for the user interacting with the toxic friend detector system.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Alamsyah, M. F. Rachman, C. S. Hudaya, R. P. Putra, A. I. Rifkyano, and F. Nurwianti, "A Progress on the Personality Measurement Model using Ontology based on Social Media Text." 2019 International Conference on Information Management and Technology (ICIMTech), 2019, doi: 10.1109/icimtech.2019.8843817.
- [2] M. Choudhary and P. K. Choudhary, "Sentiment Analysis of Text Reviewing Algorithm using Data Mining." 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), 2018, doi: 10.1109/icssit.2018.8748599.
- [3] Y. A. Solangi, Z. A. Solangi, S. Aarain, A. Abro, G. A. Mallah, and A. Shah, "Review on Natural Language Processing (NLP) and Its Toolkits for Opinion Mining and Sentiment Analysis." 2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS), 2018, doi: 10.1109/icetas.2018.8629198.
- [4] M. Vaidhya, B. Shrestha, B. Sainju, K. Khaniya, and A. Shakya, "Personality Traits Analysis from Facebook Data." 2017 21st International Computer Science and Engineering Conference (ICSEC), 2017, doi: 10.1109/icsec.2017.8443932.
- [5] Nitin Hardeniya, Jacob Perkins, Deepti Chopra, Nisheeth Joshi, "Natural Language Processing: Python and NLTK" Oreily .com <https://www.oreilly.com/library/view/natural-language-processing/9781787285101/ch02s07.html> (accessed Sept 04, 2022)
- [6] "Removing Stop Words with NLTK in Python." GeeksforGeeks, www.geeksforgeeks.org/removing-stop-words-nltk-python/. (accessed Sept 04, 2022)
- [7] "Introduction to Stemming" GeeksforGeeks, <http://www.geeksforgeeks.org/introduction-to-stemming/>. (accessed Sept 04, 2022)

BIBLIOGRAPHY

- [8] “What is stemming? - Definition from WhatIs.com” SearchEnterpriseAI <https://www.techtarget.com/searchenterpriseai/definition/stemming> (accessed Sept 04, 2022)
- [9] Lamere, P., Kwok, P., Gouvêa, E.B., Raj, B., Singh, R., Walker, W., Warmuth, M.K., and Wolf, P., "THE CMU SPHINX-4 SPEECH RECOGNITION SYSTEM." Sun Microsystems Laboratories, USA, 2001, doi: 10.1.1.406.8962
- [10] E. Booth, J. Carns, C. Kennington, and N. Rafla, "Evaluating and Improving Child-Directed Automatic Speech Recognition." pp. 6340–6345, 2020
- [11] Suresh Kumar 1 , Dr. Shivani Goel, “Enhancing Text Classification by Stochastic Optimization method and Support Vector Machine” Vol. 6 (4) , 2015, pp 3742-3745
- [12] “Scikit Learn - Support Vector Machines” tutorialspoint. https://www.tutorialspoint.com/scikit_learn/scikit_learn_support_vector_machines.htm (accessed Sept 03, 2022)
- [13] UmniySalamah , DesiRamayanti, “Implementation of Logistic Regression Algorithm for Complaint Text Classification in Indonesian Ministry of Marine and Fisheries” Volume 5 Issue 5, Sep- Oct 2018 pp 2394-2231
- [14] Sayar Ul Hassana, Jameel Ahamed, Khaleel Ahmad “ Analytics of machine learning-based algorithms for text classification” pp Volume 3, 2022, Pages 238-248
- [15] “Understanding TF-ID: A Simple Introduction”, MonkeyLearn <https://monkeylearn.com/blog/what-is-tf-idf/> (accessed Aug 03, 2021)
- [16] Zhou Tong, HaiyiZhang, “A Text Mining Research Based on LDA Topic Modelling”, The Sixth International Conference on Computer Science, Engineering and Information Technology, doi:10.5121/csit.2016.60616
- [17] WangYue, Lei Li “Sentiment Analysis using Word2vec-CNN-BiLSTM Classification”, IEEE, DOI: 10.1109/SNAMS52053.2020.9336549

BIBLIOGRAPHY

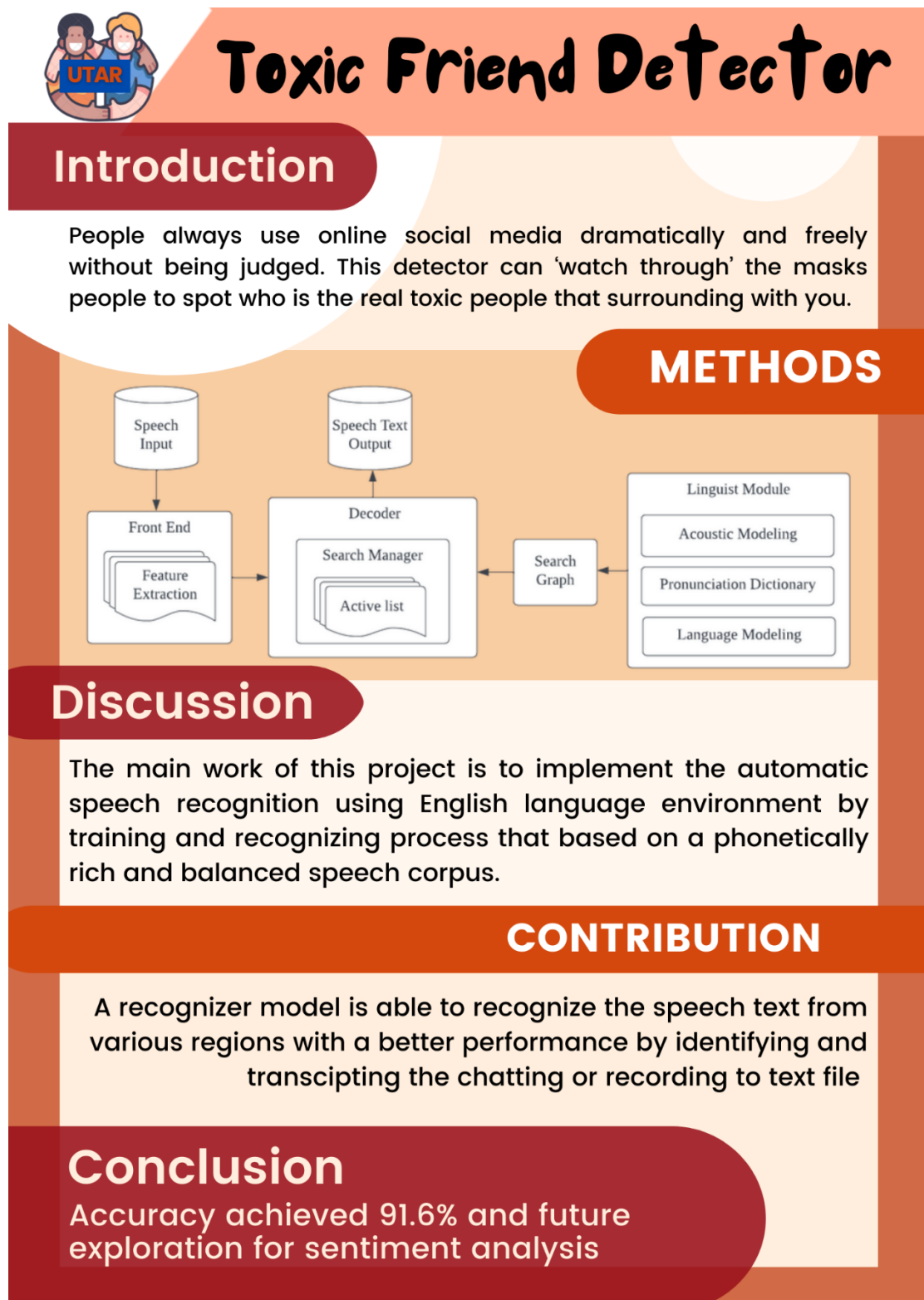
- [18] HaoTian, Liuai Wu, "Microblog Emotional Analysis Based on TF-IWF Weighted Word2vec Model", IEEE, DOI: 10.1109/ICSESS.2018.8663837
- [19] Wei Wang; Xiangshun Li; Sheng Yu " Chinese Text Keyword Extraction Based on Doc2vec And TextRank", IEEE, DOI: 10.1109/CCDC49329.2020.9164788
- [20] Hasibe Busra Dogru; Sahra Tilki; Akhtar Jamil; Alaa Ali Hameed "Deep Learning-Based Classification of News Texts Using Doc2Vec Model", IEEE, DOI: 10.1109/CAIDA51941.2021.9425290
- [21] "What is API: Definition, Types, Specifications, Documentation", altexsoft, <https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/> (accessed Sept 01, 2022)
- [22] "Python Introduction", w3school, https://www.w3schools.com/python/python_intro.asp (accessed Sept 01, 2022)
- [23] "What is Flask Python", Python Tutorial" <https://pythonbasics.org/what-is-flask-python/> (accessed Sept 01, 2022)
- [24] E. Z. Eman, "Arabic Continuous SpeechRecognition SystemusingSphinx-4", M.S. thesis, Faculty of Engineering Computer Engineering Department, 2012. [Online]. Available: <https://library.iugaza.edu.ps/thesis/105257.pdf>
- [25]D. Berrar, "Bayes' Theorem and Naive Bayes Classifier." Encyclopedia of Bioinformatics and Computational Biology, pp. 403-412, 2019, doi: 10.1016/b978-0-12-809633-8.20473-1.
- [26] S. Paulson, B. Thilagavathi, "An Adaptable Speech to Sign Language Translation System", INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) vol. 03, 2014
- [27] E. Booth, J. Carns, C. Kennington, and N. Rafla, "Evaluating and Improving Child-Directed Automatic Speech Recognition." pp. 6340–6345, 2020

BIBLIOGRAPHY

- [28] W. Walker, P. Lamere, P. Kwok., B. Raj, R. Singh, E.B. Gouvêa, P. Wolf and J. Woelfel, "Sphinx-4: a flexible open source framework for speech recognition. "" Technical Report SMLI TR2004-0811, Sun Microsystems, Inc. , 2004
- [29] M. V. C. and V. Radha, "Speaker Independent Isolated Speech Recognition System for Tamil Language using HMM." *Procedia Engineering*, vol. 30, pp. 1097-1102, 2012, doi: 10.1016/j.proeng.2012.01.968.
- [30] S. Karpagavalli and E. Chandra, "A review on automatic speech recognition architecture and approaches," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 9, no. 4, pp. 393–404, 2016.
- [31] "Building an application with sphinx4" CMUSphinx. <https://cmusphinx.github.io/wiki/tutorialsphinx4/> (accessed: Mar. 09, 2021)
- [32] Muljono, A. Q. Syadida, D. R. I. M. Setiadi and A. Setyono, "Sphinx4 for Indonesian Continuous Speech Recognition System," in *International Seminar on Application for Technology of Information and Communication*, 2017, pp. 264-267.
- [33] Lamere, P., Kwok, P., Gouvêa, E.B., Raj, B., Singh, R., Walker, W., Warmuth, M.K., and Wolf, P., "THE CMU SPHINX-4 SPEECH RECOGNITION SYSTEM." Sun Microsystems Laboratories, USA, 2001, doi: 10.1.1.406.8962
- [34] Dyuti Shukla, Mihika Shah, Prerna Parmeshwaran, and Kiran Bhowmick, "A Proposed Solution for Sentiment Analysis on Tweets to Extract Emotions from Ambiguous Statements." *International Journal of Engineering Research and*, no. 11, 2015, doi: 10.17577/ijertv4is110185

Appendix

Appendix A: Poster



Appendix C: Final Year Project Weekly Report

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	Study week no.: 2
Student Name & ID: Lai Xuan Ying & 18ACB05076	
Supervisor: Dr Aun Yichiet	
Project Title: Toxic Friend Detector	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Going through the FYP 2 guideline
- Review and recall back the previous FYP1 works and refresh memory of what have been proposed and further development.

2. WORK TO BE DONE

- Setting up the installation
- Study on the Speech recognition and Text classification modules and tutorials.

3. PROBLEMS ENCOUNTERED

- N/A

4. SELF EVALUATION OF THE PROGRESS

- Still on track



Supervisor's signature



Student's signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	Study week no.: 4
Student Name & ID: Lai Xuan Ying & 18ACB05076	
Supervisor: Dr Aun Yichiet	
Project Title: Toxic Friend Detector	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Planning Schedule
- Installation Required Software

2. WORK TO BE DONE

- Data collection
- Data pre-processing

3. PROBLEMS ENCOUNTERED

- Installation required software for beginner is a bit challenging to understand.

4. SELF EVALUATION OF THE PROGRESS

- Still on track



Supervisor's signature



Student's signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	Study week no.: 6
Student Name & ID: Lai Xuan Ying & 18ACB05076	
Supervisor: Dr Aun Yichiet	
Project Title: Toxic Friend Detector	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Data collection
- Date pre-processing
-

2. WORK TO BE DONE

- System Design
- Building model

3. PROBLEMS ENCOUNTERED

- Some functions of the recognizers are not completely installed
- Lack of source materials

4. SELF EVALUATION OF THE PROGRESS

- Still on track



Supervisor's signature



Student's signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	Study week no.: 8
Student Name & ID: Lai Xuan Ying & 18ACB05076	
Supervisor: Dr Aun Yichiet	
Project Title: Toxic Friend Detector	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- System Design
- Building model

2. WORK TO BE DONE

- Develop and testing the model
- Evaluating the performance analysis

3. PROBLEMS ENCOUNTERED

- Need more time to understand the model since many model need to be constructed into the code to enhancing the system

4. SELF EVALUATION OF THE PROGRESS

- Still on track



Supervisor's signature



Student's signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project I I)

Trimester, Year: 3, 3	Study week no.: 10
Student Name & ID: Lai Xuan Ying & 18ACB05076	
Supervisor: Dr Aun Yichiet	
Project Title: Toxic Friend Detector	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Develop and testing the model
- Evaluating the performance analysis

2. WORK TO BE DONE

- Comparing the proposed system with existing system
- Evaluation Performance analysis

3. PROBLEMS ENCOUNTERED

- Require some time to identify the existing model

4. SELF EVALUATION OF THE PROGRESS

- Still on track



Supervisor's signature



Student's signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	Study week no.: 12
Student Name & ID: Lai Xuan Ying & 18ACB05076	
Supervisor: Dr Aun Yichiet	
Project Title: Toxic Friend Detector	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Comparing the proposed system with existing system
- Evaluation Performance analysis

2. WORK TO BE DONE

- Web application development
- Connect API between front end and back end
- Writing report

3. PROBLEMS ENCOUNTERED

- Unfamiliar Flask API

4. SELF EVALUATION OF THE PROGRESS

- N/A



Supervisor's signature



Student's signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	Study week no.: 14
Student Name & ID: Lai Xuan Ying & 18ACB05076	
Supervisor: Dr Aun Yichiet	
Project Title: Toxic Friend Detector	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Web application development
- Connect API between front end and back end
- Writing report

2. WORK TO BE DONE

- Complete FYP1 report

3. PROBLEMS ENCOUNTERED

- N/A

4. SELF EVALUATION OF THE PROGRESS

- N/A



Supervisor's signature



Student's signature

Plagiarism Check Result

Appendix D: Plagiarism Check Result

Feedback Studio - Google Chrome
ev.turnitin.com/app/carta/en_us/?s=&o=1895691783&lang=en_us&u=1130770646&student_user=1

feedback studio Xuan Ying Lai Toxic Friend Detector

1.1 Problem Statement and Motivation

Information explosion is expeditiously rising in the social network, and it has become a central facilitator for daily communication with family members, peers, relatives, colleagues and friends. Naturally, it is easy to handle the unstructured data and information that lies behind texts which are discovered on social webs and data sources. These large quantities of content and text are dramatically convenient and useful for sentiment analysis. Sentiment analysis always tackles the computational prescription of subjectivity, opinion, behavior and sentiment of the text. Generally, a question of sentiment analysis always begins with “What other people think?” and this will lead to the issue of personality traits of an individual. People nowadays possess self-awareness when evaluating themselves in relation to others. However, people sometimes still are blinded and unaware of these kinds of toxic people. In order to prevent some tragic circumstances, they would like to filter and deal with these toxic people with some convenient ways. Perhaps there is a cautionary tool for predicting and identifying these texts' contents that those people who surround us are toxic or reliable-worthy.

Page: 1 of 91 Word Count: 19665

Text-Only Report | High Resolution On

Match Overview

16%

Rank	Source	Match Percentage
1	docplayer.net Internet Source	1%
2	www.ijert.org Internet Source	1%
3	mlg.anu.edu.au Internet Source	1%
4	Submitted to Universiti ... Student Paper	1%
5	Muljono, Askarya Qaul... Publication	1%
6	towardsdatascience.co... Internet Source	1%
7	"Global Trends in Infor... Publication	<1%
8	IR Putri, R Kusumaning... Publication	<1%
9	Guanqun Sun, Ao Guo, ... Publication	<1%

Toxic Friend Detector

ORIGINALITY REPORT

16%

SIMILARITY INDEX

10%

INTERNET SOURCES

9%

PUBLICATIONS

9%

STUDENT PAPERS

PRIMARY SOURCES

1

docplayer.net

Internet Source

1%

2

www.ijert.org

Internet Source

1%

3

mlg.anu.edu.au

Internet Source

1%

4

Submitted to Universiti Tunku Abdul Rahman

Student Paper

1%

5

Muljono, Askarya Qaulan Syadida, De Rosal Ignatius Moses Setiadi, Andik Setyono. "Sphinx4 for Indonesian continuous speech recognition system", 2017 International Seminar on Application for Technology of Information and Communication (iSemantic), 2017

Publication

1%

6

towardsdatascience.com

Internet Source

1%

7

"Global Trends in Information Systems and

1%



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Lai Xuan Ying
ID Number(s)	18ACB05076
Programme / Course	Bachelor of Computer Science (Honours)
Title of Final Year Project	Toxic Friend Detector

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)
Overall similarity index: <u>16</u> % Similarity by source Internet Sources: <u>10</u> % Publications: <u>9</u> % Student Papers: <u>9</u> %	
Number of individual sources listed of more than 3% similarity: <u>0</u>	
Parameters of originality required, and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Dr Aun Yichiet

Date: 9/9/2022

Signature of Co-Supervisor

Name: _____

Date: _____

Appendix E: FYP 2 CHECKLIST



UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student ID	18ACB05076
Student Name	Lai Xuan Ying
Supervisor Name	Dr Aun Yichiet

TICK (✓)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
✓	Title Page
✓	Signed Report Status Declaration Form
✓	Signed FYP Thesis Submission Form
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
✓	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
✓	Appendices (if applicable)
✓	Weekly Log
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
✓	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

(Signature of Student)

Date: 9th September 2022