



Faculty of Computer Science and Information Technology

Author Identification of English Tweets for Social Media Forensics

Nursyahirah Binti Tarmizi

**Master of Science
2023**

Author Identification of English Tweets for Social Media Forensics

Nursyahirah Binti Tarmizi

A thesis submitted

In fulfillment of the requirements for the degree of Master of Science

(Language Technologies)

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2023

DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Malaysia Sarawak. Except where due acknowledgements have been made, the work is that of the author alone. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.


.....

Signature

Name: Nursyahirah Binti Tarmizi

Matric No.: 17020134

Faculty of Computer Science and Information Technology

Universiti Malaysia Sarawak

Date: 24th August 2023

ACKNOWLEDGEMENT

In the name of Allah the most Gracious, the most Merciful, all praises to Allah S.W.T for this wonderful journey. I'm grateful for the utmost support from my family, especially my parents Harimah and Tarmizi as well as my sister Syazana, my aunts, uncles especially Masni and Hambaly, and cousins for their continuous love and du'a during the creation of my thesis. Without you I would not be where I am today.

I am very thankful to my supervisor, Dr Suhaila Saeed and co-supervisor, Dr Dayang Hanani Abang Ibrahim, who supervised me throughout my studies and always support me in my own research ideas. Even in the busiest times, feedbacks and supports were always available and the discussion of ideas helped me a lot.

I would also like to thank all my colleagues at FCSIT especially Postgraduate Lab 1, whom I have honour to change ideas and opinions on a regular basis. Besides discussing ideas and changing opinions, colleagues like them cannot be taken for granted and are among the reasons why I enjoyed this period in my life as much as I did.

My sincere gratitude to the Centre for Graduate Studies, for the advice and support given during my period of study in Universiti Malaysia Sarawak.

Finally, I would like to thank the management of the Universiti Malaysia Sarawak for making it possible for me to complete my study here in Sarawak. Thank you all.

ABSTRACT

Authorship Identification (AI) is the process of determining the most likely author of a given text by analysing writing style characteristics and linguistic patterns. Identifying the author of online social network (OSN) text becomes a pressing issue nowadays as the increase of cyberbully cases among the social media users. AI plays vital role in social media forensics (SMF) to unveil the true identity of the cyberbullying perpetrator from the OSN text. However, OSN text has been an open problem in AI as the limited length of the text and the usage of Internet jargon affecting the performance of AI system. In this research, AI task is conducted to facilitate the SMF activity by analysing the writing style of tweets from Twitter in identifying most plausible author for anonymized tweet. The writing style of the author or known as the stylometric features including character n-grams, word n-grams and Part-of-Speech (POS) n-grams are extracted from the text. These features are used widely in identifying the author of short text as they are language independent and tolerant of grammatical errors. The features are represented using different text representation models namely TF-IDF and Embedding model. The models are examined to compare which one could best represent the OSN text. For classification, machine learning and deep learning are used to evaluate the classification model by maintaining the optimum performance of AI system. The findings shown that Twitter native features are very useful in boosting the performance of AI system. Embedding-based model achieved better performance in representing n-grams with fix and distributed representation. The best result was achieved when CNN mix with embedding-based model with accuracy of 95.02% for English and 94% for KadazanDusun and both 95 % precision for both languages.

Keywords: Author Identification, authorship analysis, cyberbullying, stylometry, tweets

Identifikasi Penulis Tweets Bahasa Inggeris untuk Forensik Media Sosial

ABSTRAK

Identifikasi penulis adalah proses untuk mengenalpasti penulis yang paling munasabah bagi sesuatu text dengan menganalisa ciri gaya penulisan dan corak linguistik. Pengenalpastian identiti penulis text media sosial menjadi isu yang serius pada masa kini akibat peningkatan kes jenayah pembuli siber dalam kalangan pengguna media sosial. Identifikasi penulisan memainkan peranan yang penting dalam forensik media sosial untuk mendedah pemilik sebenar teks pembulian siber. Walau bagaimanapun, teks media sosial yang pendek merupakan masalah semasa dalam identifikasi penulisan kerana ukuran teks yang terhad dan penggunaan bahasa Internet yang tidak teratur akan menjatuhkan prestasi sistem pengenalpastian identiti penulis. Dalam kajian ini, tugas AI dijalankan bagi membantu aktiviti SMF dengan menganalisa bentuk penulisan tweets daripada Twitter bagi mengenalpasti identiti penulis tweet tanpa nama. Gaya penulisan atau dikenali sebagai ciri-ciri stylometrik, seperti n-gram abjad, n-gram perkataan dan n-gram Part-of-Speech (POS) akan diekstrak. Ciri-ciri stylometrik ini digunakan secara meluas dalam mengenalpasti penulis teks media sosial kerana ciri-cirinya yang bersifat berdikari bahasa dan dapat bertolak ansur dengan kesalahan tatabahasa. Dua model perwakilan teks, TF-IDF dan Embedding telah digunakan dan dibandingkan keduanya untuk meneliti yang mana antara dua model tersebut lebih baik dalam mewakili teks media sosial. Untuk proses klasifikasi, pendekatan machine learning dan deep learning telah digunakan untuk menilai algoritma klasifikasi yang mana satu dapat mengekalkan prestasi optimum bagi sistem pengenalpastian penulis text media sosial. Dapatan kajian menunjukkan bahawa ciri-ciri asli Twitter sangat berguna dalam meningkatkan prestasi system AI. Dapatan juga menunjukkan model berasaskan embedding lebih baik dalam mewakili ciri-ciri n-grams

dengan perwakilan tetap dan teragih. Keputusan terbaik dicapai apabila model CNN bergabung dengan gabungan embedding dengan ketepatan 95.02% untuk Bahasa Inggeris dan 94% untuk KadazanDusun dan keduanya mencapai 95% kepersisan.

Kata kunci: *Analisa identifikasi penulisan, pembulian siber, identifikasi penulisan, stilometri, tweets*

TABLE OF CONTENTS

	Page
DECLARATION	i
ACKNOWLEDGEMENT	ii
ABSTRACT	iii
<i>ABSTRAK</i>	iv
TABLE OF CONTENTS	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS	xiii
CHAPTER 1 INTRODUCTION	1
1.1 Authorship Analysis	1
1.1.1 Major Tasks in Authorship Analysis	3
1.1.2 Types of Data in Authorship Analysis	5
1.1.3 General process of Authorship Analysis	7
1.2 Research Motivation	8
1.3 Problem Statements	10
1.4 Research Questions	12
1.5 Research Objectives	12

1.6	Research Methodology	12
1.7	Research Scopes	14
1.8	Research Significant and Contributions	15
1.9	Thesis Structure	17
	CHAPTER 2 LITERATURE REVIEW	19
2.1	Digital Forensics	20
2.1.1	Challenges in Current Digital Forensics Solutions	22
2.2	Online Social Network	23
2.2.1	Online Social Network Issue	24
2.2.2	Features in Online Social Network as the Digital Evidence	26
2.2.3	Current Works to Mitigate Cyberbullying	27
2.3	Author Identification in Social Media Forensics	29
2.3.1	Introduction to Author Identification	30
2.3.2	OSN text as Digital Evidence	30
2.3.3	State-of-the-Art of Author Identification in Social Media Forensics	32
2.3.4	Evaluation in Author Identification for Social Media Forensics	46
2.4	Discussion	50
2.5	Summary	58
	CHAPTER 3 METHODOLOGY	59
3.1	The Proposed Workflow of Author Identification for Social Media Forensics	60

3.2	Text Preparation Phase	61
3.2.1	Collecting Tweets from Twitter	61
3.2.2	Pre-processing of Tweets to Standard Text	67
3.3	Text Classification Phase	72
3.3.1	Building Text Representation Model for Tweets	73
3.3.2	Classification of Tweets' Authors	83
3.4	Evaluating the Learned Models of Author Identification	91
3.4.1	Evaluation Metrics	91
3.4.2	Validation Technique	93
3.5	Summary	94
	CHAPTER 4 RESULT AND DISCUSSION	95
4.1	Tweets as Test Corpus	95
4.1.1	English Tweets	95
4.1.2	Under-resourced Language Tweets	96
4.2	Evaluation Strategies	97
4.2.1	Evaluation on the Importance of Twitter Native Features	101
4.2.2	Evaluation of Different Feature Representation Models	107
4.2.3	Evaluation of Different Classification Algorithms	118
4.3	Discussion	127
4.3.1	Utilising Twitter Native Features in Improving the Performance of AI	127

4.3.2	Dense Representation for Better Performance in AI for SMF	128
4.3.3	Maintaining Optimal Performance of Classification Algorithm in AI for SMF	130
4.4	Summary	132
CHAPTER 5 CONCLUSION		134
5.1	Conclusion	134
5.1.1	Research Contributions	135
5.2	Limitations	138
5.2.1	The Inadequacy of Language Resources for U-RL	138
5.2.2	Imbalance of Word Distribution in Tweets	138
5.2.3	Model Sparsity	139
5.3	Future Works	139
5.3.1	Conventional Approach in Processing U-RL Data	139
5.3.2	Hybrid-based Approach to Represent the Author's Writing Style	139
REFERENCES		141
APPENDICES		153

LIST OF TABLES

		Page
Table 2.1	Traditional bullying vs. Cyberbullying	25
Table 2.2	Comparison table between three different OSN platforms	26
Table 2.3	Comparison of different cyberbully detection systems	29
Table 2.4	The summary of the literature review of the short text of AI task	44
Table 2.5	Previous studies based on validation techniques and evaluation metrics	49
Table 2.6	The advantages and disadvantages of proposed solutions of prior works in regards to the stylometric features used	51
Table 2.7	The advantages and disadvantages of proposed solutions of prior works in regards to the feature representation model	54
Table 2.8	The advantages and disadvantages of proposed solutions of prior works in regards to the classifier used	56
Table 3.1	Example of tweets before and after normalisation	70
Table 3.2	Tokenisation of a tweet using <i>TweetTokenizer</i> and <i>word_tokenize</i>	71
Table 3.3	CNN architecture hyper-parameters	90
Table 3.4	Confusion Matrix	92
Table 4.1	The experimental strategies and configurations	99
Table 4.2	Hyper-parameter configuration for classification algorithms	100
Table 4.3	Example of a pre-processed tweet between three different datasets	103
Table 4.4	Accuracy results of TF-IDF and Embedding-based representation based on the level of feature type	111
Table 4.5	Precision, Recall and F-score of the algorithms for KadazanDusun tweets for different size of samples	122
Table 4.6	Precision, recall and F-score of four classification algorithms for <i>KadazanDusun</i> tweets	125

LIST OF FIGURES

	Page
Figure 1.1 A general pipeline of authorship analysis	2
Figure 1.2 Major tasks in Authorship Analysis	3
Figure 1.3 Types of data in Authorship Analysis	5
Figure 1.4 The general process of Authorship Analysis	7
Figure 1.5 The utilization of AI tool in digital forensics case	9
Figure 1.6 The overview of research problems, research questions and research objectives	11
Figure 1.7 The proposed workflow of Authorship Identification (AI) system for Social Media Forensics (SMF)	13
Figure 1.8 The overview of research contributions	16
Figure 2.1 Three aspects of study covered in this study	19
Figure 2.2 The process model of digital forensics	21
Figure 2.3 Cyberbully activities	24
Figure 2.4 The basic components of AI system for SMF	32
Figure 2.5 Types of stylometric features	35
Figure 2.6 The architecture of profile-based vs. instance-based approach	42
Figure 2.7 Confusion matrix	47
Figure 3.1 General text classification workflow (Ferreira, 2017)	59
Figure 3.2 The proposed workflow of AI for SMF	60
Figure 3.3 Crawling tweets using keywords in the feedlist	63
Figure 3.4 Twitter users with their respective number of tweets	65
Figure 3.5 Process of gathering tweets from Twitter users	
Figure 3.6 Total number of tokens and vulgar word occurrences of each user	66
Figure 3.7 Pre-processing of collected tweets	67

Figure 3.8	Cleaning the tweets from undesired data	68
Figure 3.9	Normalising the tweets into standard text	69
Figure 3.10	Tokenisation of tweets	70
Figure 3.11	POS tagging on tokenised tweets	72
Figure 3.12	Text classification phase	73
Figure 3.13	Feature extraction of N-grams in TF-IDF model	75
Figure 3.14	Embedding-based representations of tweets	79
Figure 3.15	A projection of embedding vectors of similar words using Chars2vec model (Intuition Engineering, 2019)	81
Figure 3.16	Embedding N-grams CNN architecture	88
Figure 3.17	k cross-validation technique	93
Figure 4.1	The sequence of answering the research questions	98
Figure 4.2	Experimental design for Experiment 1	102
Figure 4.3	Accuracy comparison of three different datasets for English and <i>KadazanDusun</i> tweets	104
Figure 4.4	The design of Experiment 2 strategy	108
Figure 4.5	Accuracy results of TF-IDF model for English and <i>KadazanDusun</i> tweets using SVM and CNN	112
Figure 4.6	Accuracy results of embedding-based model using CNN and SVM for English and <i>KadazanDusun</i> tweets	115
Figure 4.7	Accuracy comparison between TF-IDF and Embedding-based using SVM and CNN as classifiers	117
Figure 4.8	Experiment 3 strategy	120
Figure 4.9	Accuracy of four different algorithms with varying number of tweets for English and <i>KhadazanDusun</i> tweets	121
Figure 4.10	The accuracy results of four different algorithms with varying number of authors for English and <i>KadazanDusun</i> tweets	124
Figure 5.1	Mapping of ROs to RCs	135

LIST OF ABBREVIATIONS

AI	Author Identification
AV	Author Verification
AP	Author Profiling
CNN	Convolutional Neural Network
DFD	Digital Forensics Department
DNN	Deep Neural Network
DL	Deep Learning
ML	Machine Learning
OSN	Online Social Network
TF-IDF	Term Frequency – Index Document Frequency
TLS	Transport Layer Security
TOR	The Onion Router
SMF	Social Media Forensics
SVM	Support Vector Machines
Word2Vec	Word-to-Vector
U-RL	Under-Resourced Language

CHAPTER 1

INTRODUCTION

Chapter 1 provides an overview of authorship analysis, including the major tasks, data types, and processes involved. It also discusses the research motivation, questions, problems, objectives, methodology, significance, and contributions of the study. The chapter concludes with a summary of the thesis structure. In the rest of the thesis, the term ‘AI’ is referred as Authorship Identification task.

1.1 Authorship Analysis

Authorship analysis is a process of examining the characteristics of a piece of writing to draw conclusions on its authorship. Authorship analysis has been used in a small but diverse number of application areas including identifying authors in literature, in the program code, and in forensic analysis for criminal cases (De Vel et al., 2001). Authorship analysis has rooted in the authorship attribution problem of historical literature such as, resolving the debate on Shakespeare’s work, solving the author debates over the *Federalist Papers* and *Unabomber Manifesto* (Zheng et al., 2003). Figure 1.1 illustrates the general pipeline of authorship analysis which describes the major authorship tasks, scenarios involved, types of data used, and the general process involved.

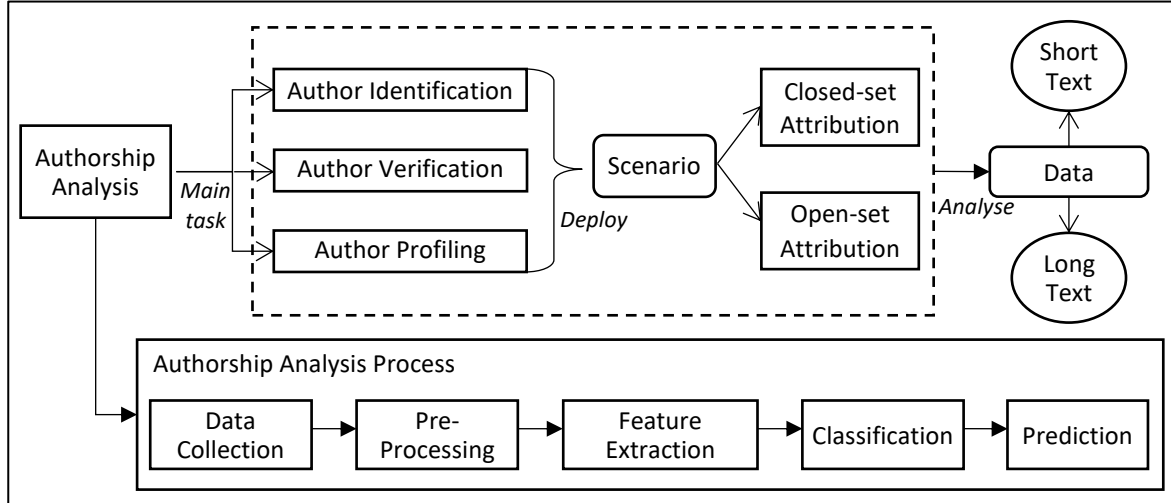


Figure 1.1: A general pipeline of authorship analysis

Referring to Figure 1.1, there are three main tasks in authorship analysis. The tasks consist of processes in determining the authorship information of the anonymous text. The three tasks namely Author Identification (AI), Author Verification (AV) and Author Profiling (AP), may be deployed in one of the two scenarios. The scenarios are called the closed-set and open-set attribution. There are two types of data that are used in gathering the internal evidence of the writing style. The general process of authorship analysis fits the standard modern paradigm of text classification problem (Koppel et al., 2009) where the components of text classification are used in authorship analysis process.

Authorship analysis involves evaluating the writing characteristics to make inferences about who wrote it. Kešelj et al. (2003) states that an author's writing style is formed by extracting stylometric features from their own texts. According to De Vel et al. (2001) stylometric features or "style markers" are used in early authorship attribution studies to define the features and quantifying the writing style of an author. In this report, the term stylometric features will be used in the explanations.

As an example, the style of an author can be represented by the choice and use of words, organisation of the sentences and paragraph, and the use of function and content words. Prior to the stylometric feature, a line of research known as ‘stylometry’ attempts to define the features for quantifying the writing style (Stamatatos, 2008). Stylometry is a linguistic discipline that applies statistical analysis to literary style (Diurdeva et al., 2016).

1.1.1 Major Tasks in Authorship Analysis

Authorship analysis task tackles the problem of determining the author of anonymous text or text that the authorship is open to dispute. Through authorship analysis task, one can solve the mystery of suicide note, ransom note or even threatening email by identifying the real culprit and release the innocent people which work close to forensic investigation and criminal law. As for the domain humanities, one would like to solve the attribution of the disputed document or reveal the anonymous author that utilising the historical documents or literary documents. There are three major tasks in authorship analysis as shown in Figure 1.2 followed by the explanation of each task.

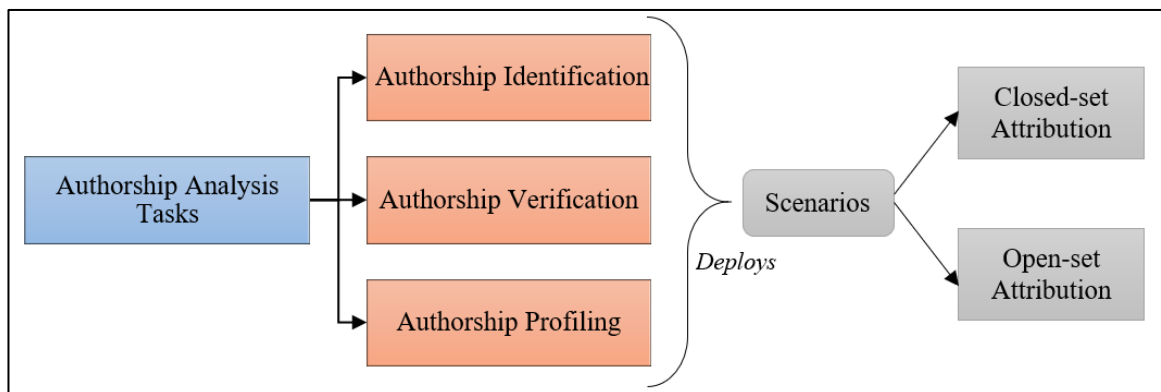


Figure 1.2: Major tasks in Authorship Analysis

Based on Figure 1.2, Authorship Identification (AI) aims to identify the author of a given document from a set of suspects by examining the other writings by that author. De Vel et al. (2001) states that the goal of AI task is to predict the author of the text selected from a pool of potential candidates. The task is called as authorship (or author) identification or authorship attribution by researchers with a background in computer science (Stamatatos, 2009).

Author Verification (AV) aims to determine whether the examined text is also written by the same author as reported by Stamatatos (2017). Rather than trying to select the most plausible author among a given number of candidates, the classification process of AV task consists of two possible outcomes either yes (indeed the person wrote the examined text) or no (the text was not written by the person).

Author Profiling (AP) is done by extracting information about the age, education, gender, etc. of the author of a given text (Koppel et al., 2002). Author profiling is only limited to determine the author's demographic including gender, age class, and native language (Soler-Company & Wanner, 2017). Besides, the profile of an author can also be the author's psychological traits (Argamon et al., 2009).

According to Figure 1.2, the authorship analysis tasks can be carried out based on two types of attribution scenarios. In the close-set scenario, a text of unknown authorship is attributed to one candidate author, given a well-defined list of candidate authors with the sample of texts they authored (Potthast et al., 2019). While, in the open-set scenario, several varieties of attribution problems fall under this scenario such as:

- There is no candidate set at all. In this case, the challenge is to provide as much as demographic or psychological information about the author.

- Needle-in-a-haystack problem happened when there are thousands of candidate authors but with limited text samples provided.
- The true author may not be included in the list of suspects and the challenge is to determine if the author is the owner of the text or otherwise.

1.1.2 Types of Data in Authorship Analysis

Figure 1.3 below depicts different types of data that are used as the evidence to be analysed in any authorship task.

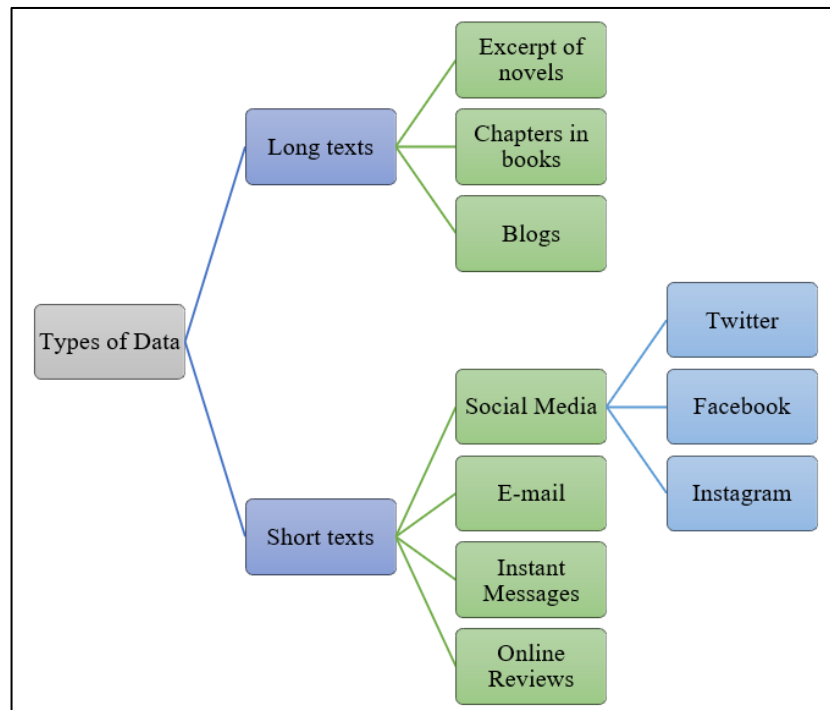


Figure 1.3: Types of data in Authorship Analysis

Based on Figure 1.3, there are two types of data that are being analysed in authorship analysis, which are short text and long text. Long texts data composed of texts from novel excerpts and books for historical literature study (Gladwin et al., 2017). However, the

extensive amount of electronic texts available through Internet media (blogs and web forum) has increased the need for handling the text efficiently. Long text has been used to analyse the writing style of terrorist groups (Abbasi & Chen, 2005) for forensic study to track the communication of militant groups and terrorist organizations using e-mails as evidence.

Presently, with the evolution of the Internet and information technology, people often use online communication to interact and socialize with each other through networks. Digital documents that are produced through the Internet are handy and essential to perform authorship analysis on short texts for the purpose of forensic investigation (Banga & Mehndiratta, 2017; Okuno et al., 2015; Orebaugh & Allnutt, 2009). Short texts such as Instant Messaging (IM), social media (Facebook, Twitter, Instagram etc.) and reviews of online products, produce massive digital documents.

Unfortunately, the evolution and divergence of the Internet are often being used negatively on online activities. Cyber-crimes such as Internet scams, stealing identity and cyberbullying (twitter, 2013) keep growing each day. According to (Diurdeva et al., 2016), the usage of digital documents processing method is crucial nowadays to analyse short text document in solving cybercrime issues.

However, AI for short text has always been difficult compared to long text. Practically, the task of extracting stylometry features easier for long texts as there are many textual features can be extracted for each author. Whereas short texts are harder to deal because of the text-length limitation and insufficient content. Thus, the limited amount of text-length and content make the identification process much harder and complicated for short text.

1.1.3 General process of Authorship Analysis

Authorship analysis has adopted the general process of text classification. In the simplest form of problem, authorship analysis task is to predict the author of disputed text given the examples of the writing of candidate authors. This makes authorship analysis task fits the standard modern paradigm of a text classification process. Authorship analysis studies differ in terms of the stylometric features used and the technique used to extract the features and classify the author's text. Figure 1.4 below shows the general process of authorship analysis.

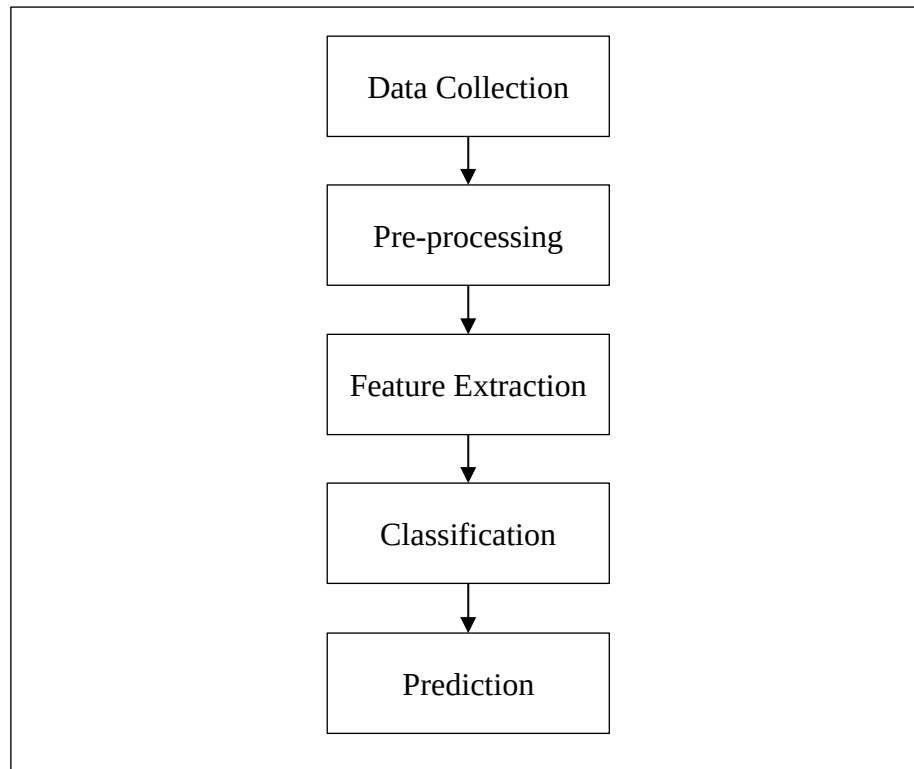


Figure 1.4: The general process of Authorship Analysis

Based on Figure 1.4, the first step of authorship analysis is data collection. Data is the text collected for each author. After collecting a sufficient amount of text for each author,

the next step is pre-processing. In pre-processing, the raw version of the text is cleaned from meta-data and analysed by using a text analysis tool to extract the textual information. Then, the textual information is extracted as stylometric features and converted into vectors. Later, the features are learned by the classification algorithms (e.g., Naïve Bayes, Neural Networks, and Decision Trees). Lastly, the learned models will be used to predict the author of anonymous text.

As stated by (Markov, 2017), from a machine learning perspective, approaches to AI task can be viewed as a multi-class single-label text classification problem. The problem includes a set of class labels which are known as prior. While, for AV task, the classification of data involves binary classification which verify the documents do belong to the author or otherwise. Similar to AV, AP consists of determining the demographic and psychological characteristics of the author of the anonymous document. For instance, to determine whether the author is male or female.

1.2 Research Motivation

Social media is increasingly popular and has become a good socialising tool where people can get connected without any limitations as of distance and times. Social media facilitates the creation and sharing of various form of content such as blogs, videos and photos. However, the usage of online activities via social media often being used negatively. Cyber-criminal activities such as social media cyberbullying keep growing each day (Bhargava et al., 2013).

Moreover, recent technology called TOR (The Onion Routing) network provides a service that encrypts user data and randomly sends back and forth to various nodes of anonymisation. For instance, one can simply tunnel traffics through a series of proxy server,

thus rendering a harder way to locate the originating IP address. The advanced network technology may create frustration on network forensics to track down the criminal where the online identity and the location of the criminal are invisible. In such case, the text left on social media may be the only clue to the author's identity.

Nowadays, there are organizations handle cyber-criminalistics investigations who reveal the identity of the anonymous user on the internet for criminal prosecution purposes. Figure 1.5 below depicts the authorship identification task in the digital forensics' investigation case.

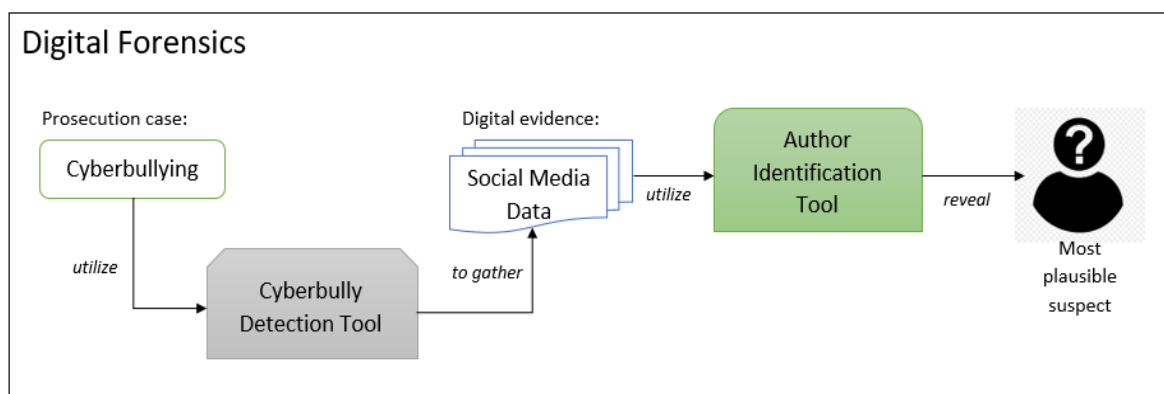


Figure 1.5: The utilization of AI tool in digital forensics case

In Figure 1.5, AI task is being carried out in digital forensics to facilitate further the prosecution by identifying the most plausible suspect that conducted the cyberbully activity in social media. Henceforth, the importance of authorship analysis in digital forensic investigation necessitates the needs for author identification task. The task of AI in social media forensics is decisive in applications of cybersecurity, social media analytic and digital humanities.

1.3 Problem Statements

Main problem: Short text issue leads to low accuracy in AI system due to a limited amount of text provided from each author (Huang et al., 2020).

Authorship attribution for short text has been a primary problem in the area of digital forensic investigation (Rocha et al., 2017). Thus, enormous training data is required due to the limited size of text samples. Therefore, the main challenge in AI for social media forensics is the analysing small number of texts from each author which later, affecting the performance of AI system (Suman et al., 2022).

Problem 1: Discriminating the authors solely on stylometric features (textual information) are insufficient and insecure as counter-attribution techniques have emerged to thwart the AI system (Theophilo et al., 2021).

In traditional authorship attribution task, discriminating the authors by extracting only the textual features are insufficient and insecure enough, as the counter-attribution techniques have emerged to thwart the AI system (Theóphilo et al., 2019). Counter-attribution is a strategy of author obfuscation (identity hiding), imitation (intentionally impersonate other author's style) and translation (using machine translation to alter an author's signal).

Therefore, instead of depending solely on the stylometric features, the meta-data contained in Twitter for instance, that contain hashtags, URLs and user references, can be utilized and embedded together in stylometric features. The meta-data or native features of Twitter may refine the accuracy of attribution by pursuing a hybrid approach that extends beyond just stylometry (Rocha, et al., 2017).

Problem 2: Complex attribution problem due to the limiting factor and informal writing style in a social media text (Fabien et al., 2020).

When dealing with Internet parlance in social media especially Twitter (e.g. emoticons, emojis, slangs, exaggerated punctuation), high usage of irregular language poses a challenge by bringing any assumption of authorial consistency on modelling the accurate system into question (Boenninghoff et al., 2019). Unlike formal text, tweets often imitate spoken language voice and mood play (Hu et al., 2020).

Authors may express the importance of words by using capital letters or repeated alphabets. Besides, excessive punctuation and Emoji are used to emphasize things and express feelings (Green & Sheppard, 2013). In order to adapt AI to social media, a suitable approach to represent the stylometric features is needed in capturing the diversity of language composed in the social media texts (Fedotova et al., 2021). Figure 1.6 illustrates the correlation between research problems, research questions and research objectives.

**Referring to Figure 1.6 and in other subsections, RQ=Research Question, RO=Research Objective, SM=Social Media*

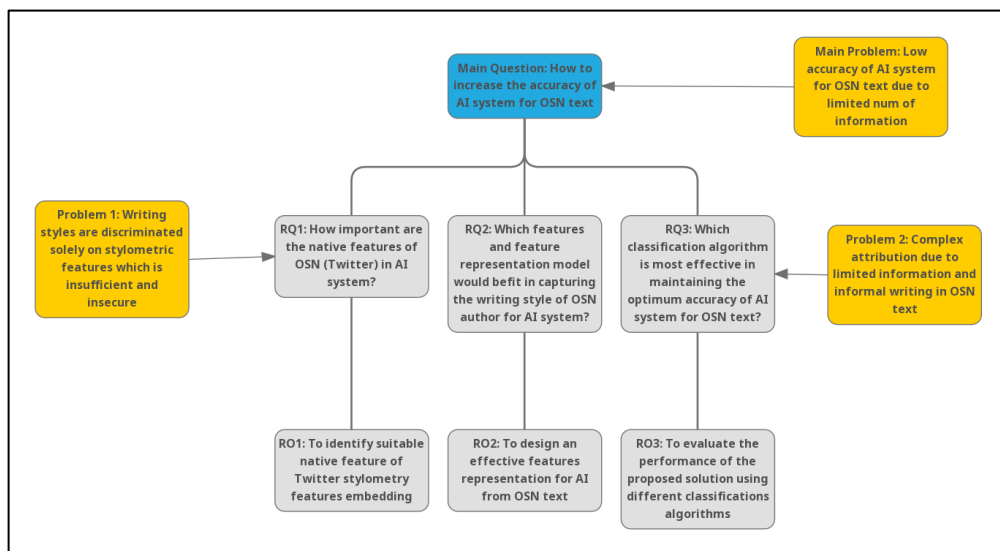


Figure 1.6: The overview of research problems, research questions and research objectives

1.4 Research Questions

As depicted in Figure 1.6, the research questions are:

Main RQ: How to increase the accuracy of AI system for OSN text?

RQ 1: How important are the native features of Twitter in AI system?

RQ 2: Which feature, and feature representation model would be most effective in capturing the writing style of the author in OSN text for AI system?

RQ 3: Which classification algorithm is most effective at maintaining the optimum accuracy of AI systems for OSN text?

1.5 Research Objectives

As a result of the research questions constructed in section 1.4, the main purpose of this research is to devise a model that incorporate suitable features from OSN and stylometric features for AI of anonymized OSN short text. The objectives of this research consist of:

RO 1: To identify suitable native features of Twitter stylometry features embedding.

RO 2: To design an effective feature representation for AI from OSN text.

RO 3: To evaluate the performance of the proposed solution using different classification algorithms.

1.6 Research Methodology

The research methodology deploys AI for social media forensics workflow as depicted in Figure 1.7. The explanation of each phase in the workflow is discussed in the following paragraph.

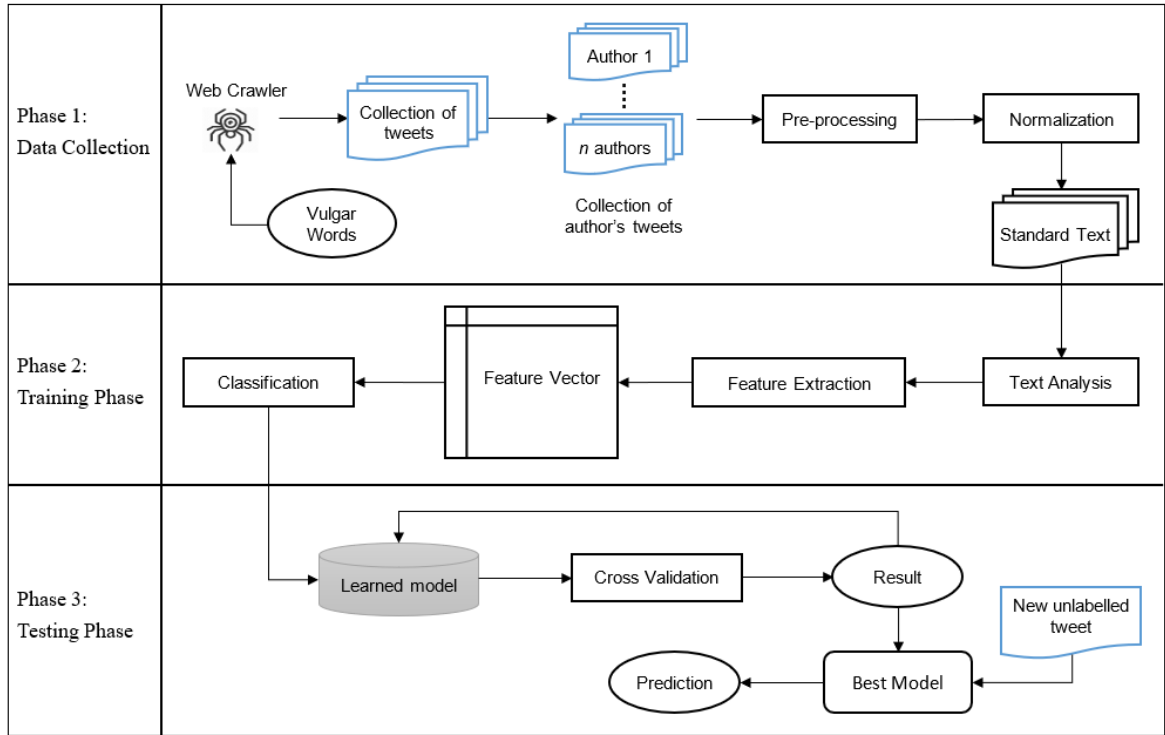


Figure 1.7: The proposed workflow of AI system for SMF

Figure 1.7 depicts the proposed workflow of AI for SMF. The workflow consists of three crucial phases which includes data collection phase, training phase and testing phase. In phase 1, a Twitter web crawler was used to crawl tweets using a set of keywords as feedlist. The remaining tweets from each author/user are collected by identifying the unique user ID. The tweets were then cleaned to remove retweets, memes, non-English tweets, and tweets with less than 3 words. The cleaned tweets were then used for training purposes.

During the training phase, text analysis tool is employed to analyze the dataset, which involves tokenization and POS tagging. Stylometric features are then extracted and transformed into feature vectors, using feature representation models. These feature extraction models are fitted into a multi-class classifier to train the models for classification.

The AI problem aims to predict the author, and a supervised machine learning approach is adopted. Machine learning and deep learning are employed to classify the feature vectors.

During testing phase, multiple models and classifiers are simultaneously validated using different numbers of authors and samples. Cross-validation is employed to evaluate the models during the validation process where the dataset is split into k groups. Each group is used as a testing set once, while the remaining groups are used as the training set $k-1$ times. This helps to prevent overfitting and allows each sample to be used for both testing and training. The best model is then selected and used to predict new, unseen tweets.

1.7 Research Scopes

The scopes of this research are:

- Focused on identifying the author of the tweet by analysing the textual information provided by the author ahead of time.
- Dataset consist of English-language tweets and Under-resourced language (U-RL) tweets, KadazanDusun, that have been collected through Twitter using a set of keywords as feedlist.
- Conducted in the closed-set scenario of AI where the prediction of the author of a new tweet is among the known authors.
- The proposed solution is based on supervised learning using machine learning and deep learning approaches.

1.8 Research Significant and Contributions

This research could provide information on the elements that are native to social media which is outside the realm of proper writing and extends the features beyond stylometry. The native elements of social media can help in combating the threat of counter-attribution such as hoaxes, imitations and identity theft. The proposed approach provides a solution for AI of a short text in social media forensics that incorporates knowledge of stylometric features and native features of social media which is difficult to tamper with.

Furthermore, this research would be beneficial to the digital forensics and cyber-crimes investigation organizations as this research would provide necessary information on the textual and non-textual data from the social media service which is important in finding the suspect of cybercriminal activities workflow that works on AI system for social media forensics . The fusion of different elements of textual and non-textual data in social media could hone the personal writing style of the author although the text provided is short and limited.

The contributions of this research are illustrated at the bottom of the chart in the following Figure 1.8.

**Referring to Figure 1.8, RQ=research question, RO=research objective, SM=social media, RC=research contribution*

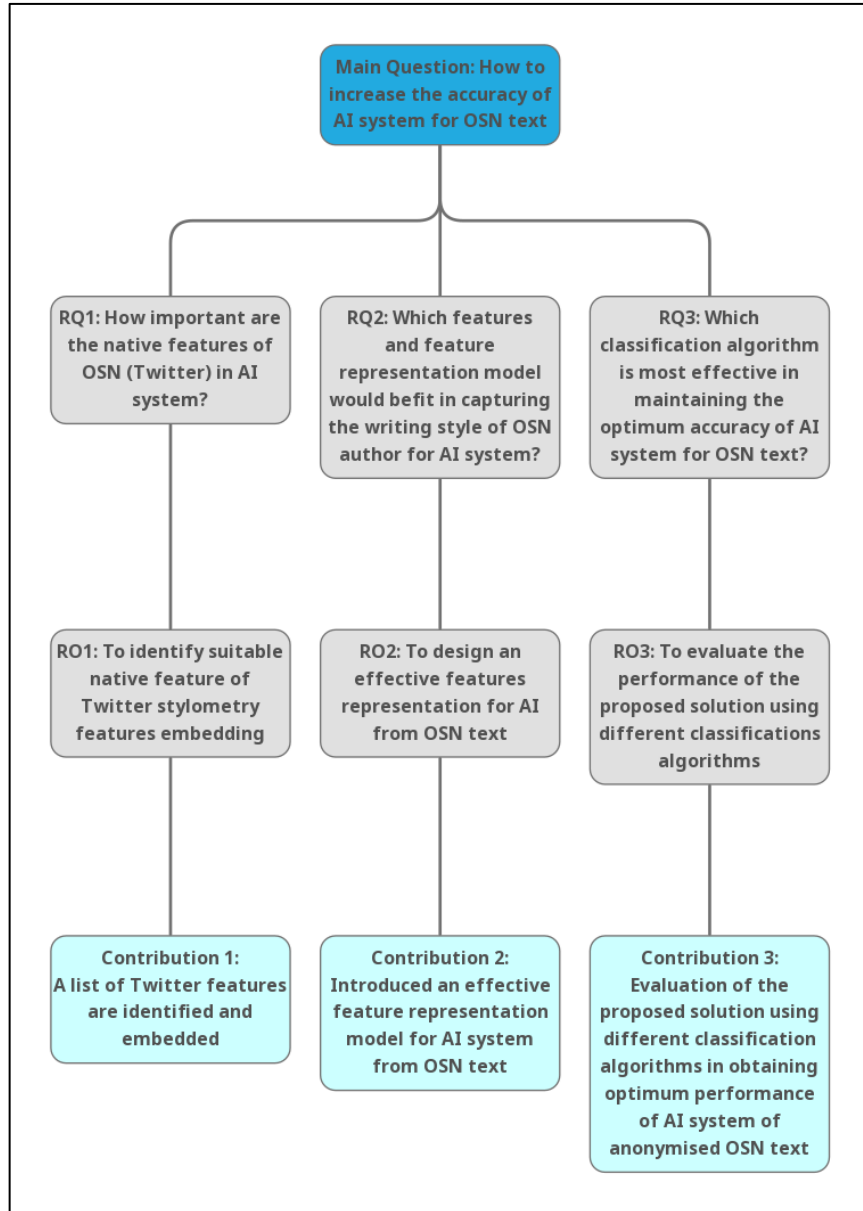


Figure 1.8: The overview of research contributions

Referring to Figure 1.8, below are the descriptions of the research contributions:

RC 1: A list of Twitter native features are identified and embedded in stylometric features.

The inclusion of native features of Twitter will hone the accuracy of AI system in predicting the suspect. Thus, by pursuing a hybrid approach that extends just beyond

stylometry will give new insight to both digital forensic and authorship fields to thwart the counter-attribution techniques that are more difficult to tamper with.

RC 2: Introduced an effective feature representation for AI system from OSN text.

Due to the special attribute imposed by the short text, usage of feature extraction model to represent the text is crucial. Therefore, in this research two different feature representation models which are TF_IDF models and embedding-based models are compared. The performance of these models is compared and analysed based on the number of author and the number of text samples.

RC 3: Evaluation of the proposed solution using different classification algorithms in obtaining optimum performance for AI of anonymized short OSN text.

Due to the increasing number of users and documents in digital form, it is important to validate state-of-the-art classifiers in AI of a short text in the context of social media or microblogging sites. Therefore, the performance of the classifiers will be evaluated using different number of authors and samples.

1.9 Thesis Structure

Chapter 1 discusses the concept of authorship analysis is introduced. In this chapter, the research overview that includes problem statements, research questions, objectives, methodology, scopes, limitations, significance, and contributions are discussed in this chapter.

Chapter 2 explores the literature review on AI task in social media forensics. Based on the previous study, the techniques, features and data are elaborated and compared.

Review AI models that are relevant in social media forensics context and the features used mostly in analysing social media text.

Chapter 3 describes the methodology of this research and how AI workflow is being implemented in social media forensics. The process in each phase is explained step-by-step in this chapter. More in-depth elaboration about the features and algorithms used will be explored and explained.

Chapter 4 elaborates the findings and analysis of the result. Comparison of the performance using different N-grams models and classifiers will be discussed in this chapter.

Chapter 5 reviews the overall conclusion of this research. In this chapter, the objectives will be evaluated. Limitations and recommendation for future works will be conferred in this chapter.

CHAPTER 2

LITERATURE REVIEW

Chapter 2 delves into the advanced techniques of Author Identification (AI) within the field of Social Media Forensics (SMF). The chapter provides a comprehensive explanation of digital forensics and its connection to AI. Figure 2.1 illustrates the interplay between digital forensics, social media, and AI, highlighting the core areas of study.

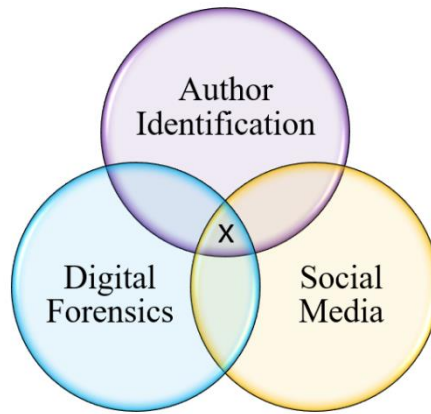


Figure 2.1: Three aspects of study covered in this study

Referring to Figure 2.1, the chapter reviews the current implementation of digital forensics in mitigating the social media issue of cyberbullying. It then explores how AI can be used to solve the limitations of the current digital forensics solution by using OSN text as evidence. The state-of-the-art AI for SMF is reviewed, and the significance and challenges of AI in analysing short OSN text for SMF are discussed.

2.1 Digital Forensics

Digital forensics is a field of computer science that deals with the recovery, preservation, and examination of digital evidence. It is used in a variety of contexts, including cybercrime investigations, civil litigation, and corporate investigations (Liu, 2009). It is a complex and challenging field, but it is becoming increasingly important as more and more crimes are being committed with the use of electronic devices (Hewling, 2009).

As stated by O'Day & Calix (2013), three different approaches are proposed by the experts in digital forensics field:

- i. Preservation, isolation, correlation, and logging: This approach involves preserving the device from being tampered with or destroyed, then examining it and logging any important findings.
- ii. Acquisition, analysis and presentation: This approach involves obtaining the device, extracting the data, and analysing it to present the findings in court.
- iii. Preservation, extraction and interpretation: This approach involves securing the device from modification or deletion, then acquiring and analyzing the digital evidence based on the scope of the investigation.

Digital forensic process models have been developed to address the specific needs of investigative workflows. These models provide structure and guidance throughout the digital forensic process, while addressing issues such as linearity, generalization, and rigidity. While there are diverse variants between the process models, there are still essential commonalities in how certain stages are used across multiple models (Sachowski, 2019). These commonalities are summarized in a workflow diagram, as illustrated in Figure 2.2.

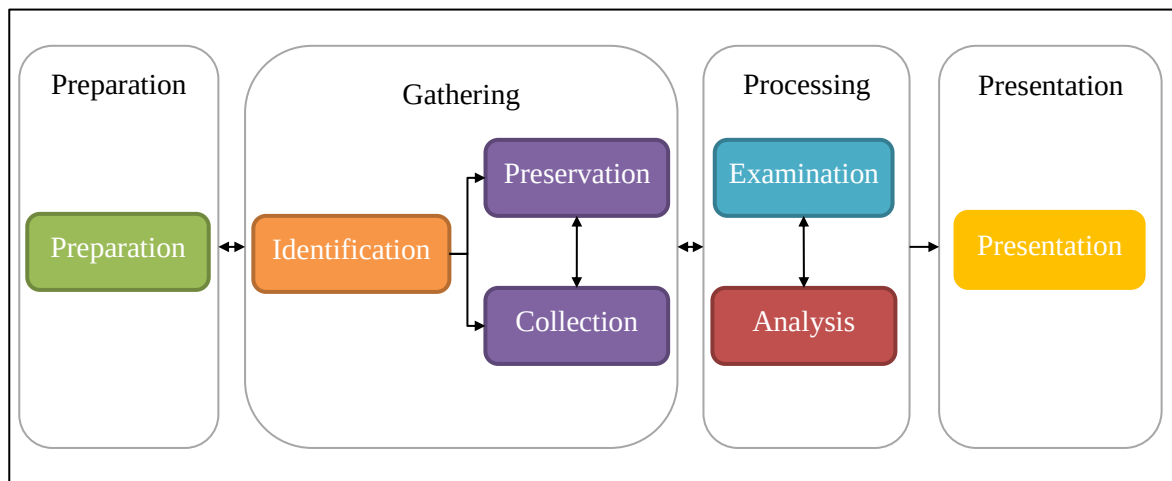


Figure 2.2: The process model of digital forensics

Figure 2.2 illustrates the digital forensic process, beginning with preparation to ensure equipment and personnel readiness. The identification step involves detecting the incident, followed by data collection for evidence. Preservation is then performed to securely gather evidence. The examination process evaluates and reduces the volume of evidence, while the analysis step examines the content and context for relevance. Finally, the presentation step involves reporting and documenting the findings.

Digital forensics encompasses various sub-disciplines, including computer forensics, network forensics, mobile device forensics, digital image forensics, digital video/audio forensics, and memory forensics. In Malaysia, the Digital Forensics Department (DFD) offers digital forensics investigations and examinations, specifically in audio and video forensics. Their clients include the Law Enforcement Agencies (LEA), Government-Linked Companies (GLC), and private agencies. DFD provides several services based on the type of medium involved in the case, such as:

- Computer forensics: perform a scientific examination and data analysis on computer storage media
- Mobile phone forensics: perform a scientific examination and data analysis on mobile phone devices
- Audio forensics: perform digital audio science and technology on digital audio files or media
- Video forensics: perform digital video science and technology digital video file

In addition to gathering offline digital evidence captured in digital devices, law enforcement and academia have shown a growing interest in data gathered from OSN. These networks, such as Facebook, Twitter, LinkedIn, and Instagram, are heavily used to socialize and keep connected, or to gain insight into people's behavior and activities. Each day, terabytes of data are generated over these networks, and this data can be used by businesses to generate revenue or by criminals to exploit vulnerabilities in the networks. As a result, digital forensics investigators are increasingly turning to online social networks to gather evidence in criminal cases.

2.1.1 Challenges in Current Digital Forensics Solutions

Digital forensics has gained significant attention due to the increase in crimes involving electronic devices (Ngejane et al., 2021). It plays a crucial role in gathering vital evidence for law enforcement and the courts. The traditional approach to digital forensics is to seize a suspect's device and bring it to a forensic lab for analysis (Mulazzani, 2014). However, this approach has two main limitations (Hamad & Eleyan, 2022):

- i. The device must be accessible to the investigator. This can be difficult if the device is encrypted or if the suspect has destroyed the evidence.

- ii. Data must be stored, shared, and communicated in a secure manner. This can be a challenge if the data is large or if the communication channels are not secure.

In addition, cybercriminals are increasingly using counter-forensic techniques to make it more difficult for investigators to gather evidence. For example, they may encrypt their communications or use anonymizing services like TOR. TOR is a popular anonymizing service that allows users to browse the internet anonymously. It does this by routing traffic through a series of servers, making it difficult to track the source of the traffic.

The anonymity provided by anti-spying technology presents an opportunity for cyber-criminals to exploit the cyber world's anonymity for illegal activities. The perception of anonymity in online activities makes cyber-criminals feel safe and secure, further complicating the situation for law enforcement agencies in their efforts to track down cybercriminals.

The digital forensics field is a complex and ever-changing field, but it is essential for investigators to stay up-to-date on the latest trends and techniques. By doing so, they can ensure that they are able to collect and analyse digital evidence effectively and efficiently.

2.2 Online Social Network

The widespread use of the Internet and OSN like Facebook, Twitter, Instagram, and Snapchat has become an integral part of people's daily lives for online interaction. However, this evolution has also led to the misuse of these platforms for illicit activities, attracting cybercriminals to engage in cyberbullying, fake news dissemination, and even child exploitation (Kaikini et al., 2022). The enormous amount of data generated on social media offers the potential for leveraging digital evidence to address these issues.

This section explores the impact of social media on digital forensics, focusing on the effects of cyberbullying on the victims, the features of OSN as digital evidence, and an overview of current efforts in digital forensics to mitigate cyberbullying in OSN. This section focuses on the impact of social media on digital forensics. The aim is to address the challenges posed by social media platforms and leverage digital forensic techniques to combat cyberbullying effectively.

2.2.1 Online Social Network Issue

Cyberbullying is a form of cybercrime (Goni et al., 2022) that involves aggressive and intentional actions toward a defenceless person using the internet or other electronic content (Park et al., 2021). According to Han et al. (2021), there are six types of cyberbullying activities, as depicted in Figure 2.3.

Cyber Harassment
•Hate speech, sexual content, self-harm, and being mean to a person
Denigration
•Unfairly critisizing someone by spreading cruel gossip or rumours about someone
Cyber Stalking
•Make false accusations, monitoring, threats, identity theft, or create data destruction and manipulaton from others information
Trolling
•Posting provocative message with intention to create conflict and upsetting others
Flaming
•Intense argument using vulgar words to provoke and offense others
Masquerade
•Create fake profile ID to keep bullying by getting one information and keep harassing

Figure 2.3: Cyberbully activities

Figure 2.3 illustrates six common types of cyberbullying in online social networks, involving the use of aggressive language to harm victims. Cyberbullying predominantly affects children and adolescents, leading to mental health issues such as depression and low self-esteem. These victims may even be driven to contemplate suicide. In Malaysia, 70% of schoolchildren have experienced cyberbullying via obscene photos, texts and being called with foul names (Ariffin et al., 2021).

Cyberbullying differs from traditional bullying in several ways. First, it can be done anonymously, which makes it more difficult to track down the bullies. Second, it can be done 24/7, which means that victims are never safe from being harassed. Third, it can be spread to a wider audience, which can make the victim feel even more isolated and humiliated. Table 2.1 presents a comparison between cyberbullying and traditional bullying, shedding light on why cyberbullies prefer social media as a medium.

Table 2.1: Traditional bullying vs. Cyberbullying

Traditional bullying	Cyberbullying
Face-to-face	Online
The bully is hard to escape	The bully can easily hide via anonymity
The scene is limited to onlookers	The scene can be shared by a broad audience (online users)
The bully can be easily identified	The bully can be anonymous
Facial and body reaction of the victim is visible	Hard to empathize with the victim

Table 2.1 highlights that cyberbullying occurs in an online setting, allowing bullies to take advantage of the anonymity it provides. This anonymity creates a false sense of

security, enabling perpetrators to hide their true identity and engage in cyberbullying with impunity, despite the potential for serious legal consequences.

2.2.2 Features in Online Social Network as the Digital Evidence

The emergence of OSN has had a profound impact on society. OSN platforms have revolutionized how people connect, access information, and engage in various activities. As the user base expands, the volume and importance of data generated on these platforms have increased significantly. This data can be valuable evidence in legal cases, as it can be used to track cyberbullying, unveil conspiracies, and identify connections between individuals.

Courts are increasingly accepting digital evidence obtained from OSN (Montasari et al., 2019). Investigations in social media forensics employ various methods to acquire this evidence, such as tracking publicly viewable content, analyzing metadata, and examining user interactions. Understanding the unique attributes of each OSN platform is essential for effectively leveraging the information provided. For example, Facebook is a more private platform than Twitter, so the data available on Facebook may be more revealing of an individual's personal life. Table 2.2 shows an example of different platforms of OSN and their distinct features.

Table 2.2: Comparison table between three different OSN platforms

	Twitter	Facebook	LinkedIn
Tone of Voice	thought provocative / informative / practical	friendly / chatty / vibrant	authoritative / informative / intelligent
Frequency of posts	timely – experiment with the time of the day for receiving most responses	regular weekly routine post + 1-2 posts a day (if relevant to topical issues & content to be posted)	once a week (until buildup 100+ followers) and then up to a frequency

Table 2.2 continued

Style of Content	include links; entice clicks by offering snippets of content for further reading	shares images & videos include links	include links; add questions; comment on conversations
Native language	@ / # /retweets / shorten URLs	@ likes / interests if relevant; tag location if appropriate	can edit the default content & image of a posted link
Character Limits	240 characters	60,000 characters	600 characters

Al-Ajlan and Ykhlef (2018) found that data from OSN can be used to track down cybercriminals and confirm alibis in criminal cases. This includes gathering meta-data, which encompasses user profiles and information stored within the OSN platform, as well as textual data i.e., posts and messages posted by users can also be collected and analysed as potential evidence. These various types of data from OSN are valuable in social media investigations, aiding in the identification and tracking of cyber-criminals.

2.2.3 Current Works to Mitigate Cyberbullying

Cyberbullying aims to harm, disrepute or embarrass a victim through 'repeated' acts such as posting inappropriate messages or spreading rumours about the victim online. Henceforth, cyber-bullies hold power to embarrass and terrorise the victims before an entire community online. Despite the anonymity nuisance, advanced machine learning solutions to surpass the cyberbully activities on OSN have increased recently.

For example, Al-Ajlan and Ykhlef (2018) presented a cyberbully detection system using a deep learning model that could eliminate feature engineering while generating better performance. They proposed a CNN model using a novel mixture of word embedding and

semantics features to develop efficient detection with accurate results. The proposed model outperformed the traditional content-based model with accuracy of 95%.

Pawar et al. (2018) developed a real-time automated tool to block cyberbullying in chat. The tool uses a bullying detection server to process incoming encrypted chat and convert it into a Bag-of-Word model. The model is then learned by classification algorithms, such as Multinomial Naïve Bayes and Stochastic Gradient Descent. The authors proposed synthetic data generation to overcome data imbalance, which improved the F1 score to 90%. Cheng et al. (2019) proposed a deep learning model to detect cyberbullying on Instagram. The model uses RNN with Bi-directional GRU to capture temporal information and encode the sequence of words and comments. The model outperformed other machine learning classifiers with an F1-score of 0.851.

Perera and Fernando (2021) proposed a new hypothesis for cyberbullying detection. They argued that bullies would use extreme words to cause harm to the victim, and that sarcastic text should not be classified as cyberbullying. They used TF-IDF, sentiment analysis, and profanity features with SVM as the classifier to classify tweets. Their findings suggest that using a combination of features can improve the accuracy of cyberbullying detection systems. All of the studies mentioned above are summarized in Table 2.3. The table covered the background of cyberbullying and the proposed cyberbully detection systems to thwart cyberbullying activities in OSN.

Table 2.3: Comparison of different cyberbully detection systems

Source	Features	Classifier	Corpus
Pawar et al. (2018)	• Textual data • BoW	• Multinomial Naïve Bayes • Stochastic Gradient Descent	Form-Spring
Al-Ajlan & Ykhlef (2018)	• Textual data • Word embedding	CNN	Twitter
Cheng et al. (2019)	• Profane words • Comments	RNN	Instagram
Balakrishnan et al. (2020)	• Application data • User personality • Sentiment • Emotion	• Naïve Bayes • Random Forest • J48	Twitter
Perera & Fernando (2021)	• Words TF-IDF • Sentiment • Profane words	SVM	Twitter

Prior researches in Table 2.3 have shown that machine learning and deep learning models can process OSN textual data to combat cyberbullying. However, perpetrators can still be anonymous. Authorship analysis can be used to identify them and facilitate forensic investigations.

2.3 Author Identification in Social Media Forensics

This section reviews the introduction of AI in digital forensics and its application in social media forensics (SMF). It discusses the types of text in social media and the importance of text as digital evidence. The state-of-the-art of AI in SMF is reviewed based on the basic components of AI. The section also discusses the evaluation and validation techniques used in AI, as well as the significance of AI for SMF and its challenges.

2.3.1 Introduction to Author Identification

AI is a process that aims to identify the author of an anonymous text by comparing it to samples of known authors (Juola, 2008). AI is a type of machine learning problem called multi-class classification, where the goal is to assign a text to one of a set of predefined authors. AI has a long history and has been used in a variety of fields, including humanities, politics, journalism, and law. The unique writing style of an author is assumed to be distinguishable from others (Soler-Company & Wanner, 2017).

The rise of the internet and OSN has created new opportunities for cybercrime activities, such as cyberbullying. The anonymity of the OSN platforms makes it difficult for law enforcement agencies to track down perpetrators. However, AI has the potential to be used in digital forensics to identify criminals and solve cases involving electronic texts. The abundance of electronic texts produced in OSN has unlocked significant potential for AI in the field of digital forensics (Stamatatos, 2008).

For example, AI techniques have been employed in intelligence operations to identify messages or proclamations linked to known terrorists in digital forensics investigations (Abbasi & Chen, 2005). AI can help solve mysteries surrounding suicide notes, ransom notes, and threatening emails by identifying the true culprits and exonerating innocent individuals. AI has also found applications in criminal law.

2.3.2 OSN text as Digital Evidence

According to Banga and Mehndiratta (2017), digital documents can serve as evidence to establish the guilt of a suspect involved in cyberbullying. Rocha et al. (2017) emphasize that the task of AI assumes that writers have unique habits that influence their writing style, including the form and content of their texts. The concept of "writeprint" is

introduced as a method to identify the author of an unknown text, thereby revealing the perpetrator of cyberbullying.

Writeprint is a form of "print" in the realm of cybercrime, which is embedded within an individual's writing. It consists of stylometric features such as vocabulary richness, sentence length, and keywords present in the text. These stylometric features quantitatively measure the writing style of an author by analysing textual characteristics, enabling differentiation between texts written by different authors.

The text generated in OSN is important for identifying cyberbullying perpetrators. However, the unique characteristics of OSN text, such as limited length, emoji, and cyber-language, make it challenging to develop a reliable AI system for analysis. Here are some of the specific challenges that AI systems face when analysing OSN text:

- Limited-length: OSN text is often limited in length, which can make it difficult to identify the author's unique writing style.
- Emoji: Emoji are often used in OSN text, which can further complicate the identification of the author's writing style.
- Overemphasize punctuation: OSN text often uses excessive punctuation marks, which can also make it difficult to identify the author's writing style.
- Cyber-language: Cyber-language is often used in OSN text, which can further complicate the identification of the author's writing style.
- Spelling errors and ungrammatical words: OSN text often contains spelling errors and ungrammatical words, which can also make it difficult to identify the author's writing style.

Despite these challenges, there is still potential for developing reliable authorship identification systems for OSN text. By carefully considering the unique characteristics of OSN text, researchers can develop systems that are more accurate at identifying the authors of cyberbullying posts.

2.3.3 State-of-the-Art of Author Identification in Social Media Forensics

Traditional methods of identifying cyberbullying perpetrators, such as interviewing witnesses and gathering physical evidence, are often not enough. New cyber forensics methods are needed to collect digital evidence, such as text messages, emails, and social media posts, that can be used to identify perpetrators.

AI system can be used to compare the writing style of a suspect to the writing style of a cyberbullying victim. If the writing styles are similar, it can provide evidence that the suspect is the perpetrator. Several components are involved in building an AI system. Figure 2.4 depicts the basic components of an AI system.

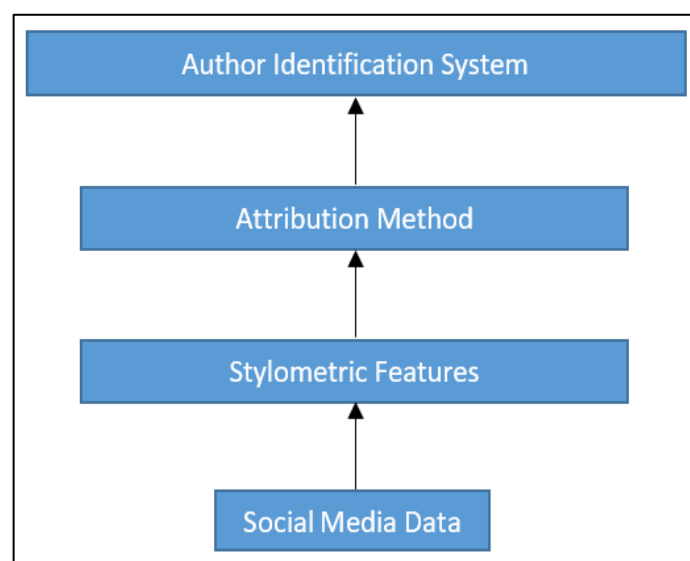


Figure 2.4: The basic components of AI system for SMF

Based on Figure 2.4, the type of input text can have a big impact on the effectiveness of authorship analysis methods. For example, lexical features are beneficial for long literature text because they can capture the vocabulary richness and function words that are commonly used in long documents (Jafariakinabad & Hua, 2019).

While, character n-grams are effective for attributing the author of OSN text because they can capture the usage of punctuation and capital letters that are commonly used in OSN text. But, character n-grams can suffer from sparse representation because they only consider the frequency of individual characters, which can lead to a lot of missing data (Posadas-Durán et al., 2017). To tackle the sparsity issue, embedding models provide dense and distributed representation by consider the relationships between different characters which can help to improve the accuracy of AI system (Hu et al., 2020).

While, the attribution method is the basic property of a classification model because it determines how the training text is treated (Kourtis & Stamatato, 2011). There are two main approaches to attribution: profile-based and instance-based. Profile-based approaches concatenate all the training text in one single file per author, while instance-based approaches treat the training text one at a time per author.

a. Twitter as Testbed

AI is a key area for the application of technologies in SMF. Tweets are a popular form of OSN content that is used in cyberbullying incidents (Frommholz et al., 2016). Twitter allows tweets and other data to be collected smoothly using the Twitter API. However, there are no public datasets for Twitter that are available for authorship identification tasks. Most researchers created their datasets by scrapping tweets based on their designated keywords and settings (Rocha et al., 2017).

Doultani and Vijayalakshmi (2019) downloaded 40,000 tweets from 100 authors from 2001 onwards using the Twitter API. Rocha et al. crawled English tweets using English function words as keywords. They obtained tweets of English-speaking public users and collected around ten million tweets from 10,000 users in 6 months in 2014.

Banga and Mehndiratta (2017) collected the most recent tweets from 10 prolific Hindi authors using the Twitter streaming API. The API function restricts up to 3200 recent tweets, including retweets. Altakrori et al. (2018) crawled random tweets written in Arabic and Urdu, which has similar characters to Arabic. Random tweets from 155 different usernames were obtained. Then, from each username, they acquire a collection of additional tweets from the user's timeline with an average number of 747 tweets per username.

b. Stylometric Features

Stylometric features are extracted to construct each person's writing style. These features include the choice and use of words, the organization of sentences and paragraphs, and the distribution of function words. Natural language processing (NLP) tasks are used to extract these features. For example, the calculation of word frequencies requires a tokenizer, stemmer, and lemmatizer, while POS tagging requires a tokenizer, sentence splitter, and POS tagger. A study by Stamatatos (2009) presented a review of stylometric features for electronic text or digital documents and stated that there are five types of stylometric features to quantify the authors' writing styles. Figure 2.5 depicts the types of stylometric features.

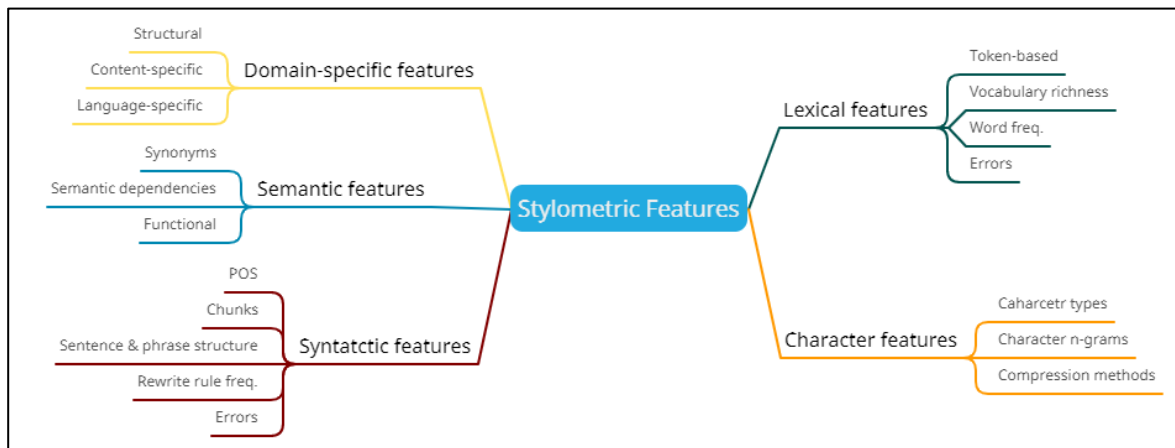


Figure 2.5: Types of stylometric features

Referring to Figure 2.5:

- **Lexical features:** These features are based on the words and phrases that an author uses. They can include the frequency of different words, the use of rare words, and the use of collocations (pairs or groups of words that often occur together) (Stamatatos, 2009; Sari et al., 2018) .
- **Character features:** These features are based on the individual characters that an author uses. They can include the frequency of different characters, the use of punctuation marks, and the use of capitalization (Chowdhury et al., 2019).
- **Domain-specific features:** These features are based on the specific domain or genre of writing that an author produces. They can include the use of technical terms, the use of jargon, and the use of discourse markers (words or phrases that signal the structure of a text) (Rivera-Soto et al., 2021).

- Semantic features: These features are based on the meaning of the words that an author uses. They can include the use of synonyms, the use of antonyms, and the use of hyponyms (words that are more specific than other words) (Hriez & Awajan, 2020).
- Syntactic features: These features are based on the structure of the sentences that an author uses. They can include the length of sentences, the use of subordinate clauses, Part-of-Speech (POS) tag information, and syntax tree (Jambi et al., 2021).

For short text like OSN text, character-level n-grams have proven to be more effective (Altakrori et al., 2018; Jambi et al., 2021) than other stylometric features. This is because character-level n-grams can capture lexical and syntactic information, as well as punctuation and capitalization information, which is important for short text. Altakrori et al. (2018) found that character n-grams were the best performing features for Arabic tweets, while word n-grams were the best performing features for English tweets. This is because Arabic words tend to merge pronouns with words, which makes it more difficult to match n-grams in Arabic.

Fourkioti et al. (2019) found that most frequent content words were the best performing features for English tweets, while combining all features showed the best accuracy. This is because English words are more distinct from each other, which makes it easier to match n-grams. Jambi et al. (2021) found that character-based features were the best performing features for Arabic tweets, while a combination of character and word-based features showed the best accuracy.

After the features are extracted from the writing samples, they are represented in the form of vector representation to be learned by the classifier for the classification process later. In machine learning, a representation model is essential because the classifiers require the text input to be represented as a fixed-length vector. The conventional way of representing text features are Bag-of-words (BoW) and term frequency-inverse document frequency (TF-IDF) model due to their simplicity, efficiency, and often surprising accuracy.

In the BoW model, a text is represented as the bag of its words based on statistics calculated from the text, disregarding grammar and word order. While, TF-IDF weighs up the important words in the corpus that helps to capture the discriminative power of terms and reduce the influence of common words which can improve the accuracy of AI.

c. Text Representations

The conventional models for representing text are the Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) models, which are popular due to their simplicity, efficiency, and often impressive accuracy. In the BoW model, a text is represented as a collection of its words, based on statistical calculations derived from the text, without considering grammar or word order (Rathod and Joshi, 2016).

On the other hand, TF-IDF assigns weights to important words in the corpus, allowing for the capture of the distinctive characteristics of terms while reducing the impact of common words. This approach proves more effective than the BoW model. However, TF-IDF with n-grams generates sparse vectors and offers limited semantic interpretation, even though it remains the most accurate style marker for AI when applied to short text (Posadas-Durán et al., 2017).

Modern approaches to representing words as vectors involve word embedding models, which allow words to be represented based on their relationships with neighbouring elements (Posadas-Durán et al., 2017). These models aim to capture semantic information and contextual relationships between words, offering improved performance in various natural language processing tasks. Word embeddings are typically represented in a low-dimensional space, such as a 100-dimensional space which allows the model to learn the relationships between words without being overwhelmed by the large number of words in the corpus.

Word embedding models have been shown to be effective for AI tasks. These models learn word embeddings by predicting the context of words or the co-occurrence of words. Some of the most popular word embedding models for author identification include Word2vec (Mikolov et al., 2013), GloVe (Pennington, 2014), FastText (Joulin et al., 2016), BERT (Devlin et al., 2018), and RoBERTa (Liu et al., 2019). These models have been shown to achieve state-of-the-art results on a variety of AI datasets.

Mahajan et al. (2018) introduced a technique that incorporates character n-grams to enhance the Word2Vec model, resulting in better performance compared to the regular Word2Vec models. Similarly, Zhang et al. (2018) combined the Word2Vec model with a syntax encoding approach, utilizing syntax parse trees to create embedding vectors. This combined model achieved superior accuracy in sentiment analysis tasks.

Chowdhury et al. (2019) implemented deep neural network-based models using fastText's continuous skip-gram model to represent word n-grams, outperforming traditional models in analyzing Bengali literature. In another study, Khatun et al. (2019) employed character 3-grams and pre-trained embedding models to achieve high accuracy in an AI task.

on Bangla text. These investigations collectively demonstrate that combining Word2Vec with n-grams can significantly enhance the accuracy of AI systems for short text.

Fabien et al. (2020) present a BERT-based approach for author identification by fine-tuning a pre-trained BERT model on labeled text data. They evaluate the method on two AI datasets, PAN 2017 and PAN 2018, achieving accuracies of 96.5% and 95.4% respectively. The authors emphasize BERT's robustness to noise and outliers, considering it a powerful language model. They highlight the importance of fine-tuning BERT on a large labeled dataset for achieving high performance on AI tasks, while acknowledging the computational cost and requirement of substantial author text data.

Huang et al. (2021) propose a novel approach to AI task that combines the RoBERTa model and TF-IDF. The hybrid model uses the fine-tuned RoBERTa model to capture author-specific writing patterns, while TF-IDF extracts distinctive features from the text. Experimental results show that the hybrid model outperforms individual models and traditional methods such as BoW and TF-IDF alone with 97.2% accuracy. The study highlights the complementary nature of RoBERTa and TF-IDF, making the hybrid model a promising solution for author identification tasks in various domains. However, RoBERTa is a more computationally expensive model and more complex than BERT.

d. Classification Algorithms

Various AI models make use of SVM classifiers to classify different stylometric features. These works include (Altakrori, et al., 2018; Fourkioti et al., 2019; Jambi et al., 2021). Among all the works, SVM outperforms other classifiers with character n-grams as the features or by combining with other features. Apart from SVM, Random Forest yielded

the best accuracy in attributing authors of the Vietnam online forum (Khanh & Vorobeva, 2020) and Arabic tweets (Altakrori et al., 2018).

Researchers have increasingly relied on neural networks in recent years to learn and classify features in text classification tasks such as topic classification and sentiment analysis.. Shrestha et al. (2017) presented a CNN model for authorship classification based on character n-grams. The model achieved good performance, improving 6% from SVM model.

Theóphilo et al. (2019) used CNN to gauge tweets on fake news issues. They achieved good performance during training, but validation accuracy dropped to 65%. They improved the model using data augmentation, achieving 74.15% accuracy. Huang et al. (2020) introduced a Bi-LSTM CNN model to classify authors of tweets. They used skip-gram model to pre-train word and character n-grams embeddings, and computed polarity and subjectivity scores to extract sentiment information. Their model achieved good results compared to Schwartz et al. (2013) and CNN model without latent features.

Punchihewa et al. (2021) chose RNN to train word and character embedding on the Sinhala news articles dataset. They obtained good results in character-level model with the highest accuracy, which achieved 56.06% using the highest volume of training samples with 100 articles. Barlas & Stamatatos (2021) demonstrated character-level RNN in cross-topics authorship attribution condition where the constraint differs from training text samples. They used the CMCC corpus, which contains English graduates' writings covering six genres and six topics. Pre-trained language model, BERT, achieved the best results across all cross-domains compared to others i.e. GPT 2, ELMO and ULMFiT.

Punchihewa et al. (2021) chose RNN to train word and character embedding on the Sinhala news articles dataset. They obtained good results in character-level model with the highest accuracy, which achieved 56.06% using the highest volume of training samples with 100 articles. Barlas & Stamatatos (2021) demonstrated character-level RNN in cross-topics authorship attribution condition where the constraint differs from training text samples. They used the CMCC corpus, which contains English graduates' writings covering six genres and six topics. Different pre-trained language models, including BERT, GPT-2, ELMO and ULMFiT are used to pre-train the embedding where BERT achieved the best results across all cross-domains.

Talati et al. (2020) proposes a novel hybrid model that combines the strengths of CNN and RNN. The CNN is used to extract local features from the text, while the RNN is used to capture long-range dependencies. The models are trained on large datasets of text samples, allowing them to learn intricate patterns and features

e. The Attribution Method

In every AI problem, there are candidate authors with their collection of text samples as the training dataset. Together with a set of test samples of unknown authorship as the test dataset, whereby the unknown authorship text should be attributed to a candidate author.

According to Stamatatos (2009), there are three approaches to extracting the writing style of each candidate author. The first is the profile-based approach. This approach extracts the writing style of an author based on the general writing style by combining the writing samples of that author.

Second is the instance-based approach. In this approach, the writing samples of an author are treated individually, and the writing style is extracted separately. The third is the hybrid approach, where some elements from both profile-based and instance-based approach are combined. Referring to Figure 2.6, building an AI system depends on how the attribution model handles the stylometric features.

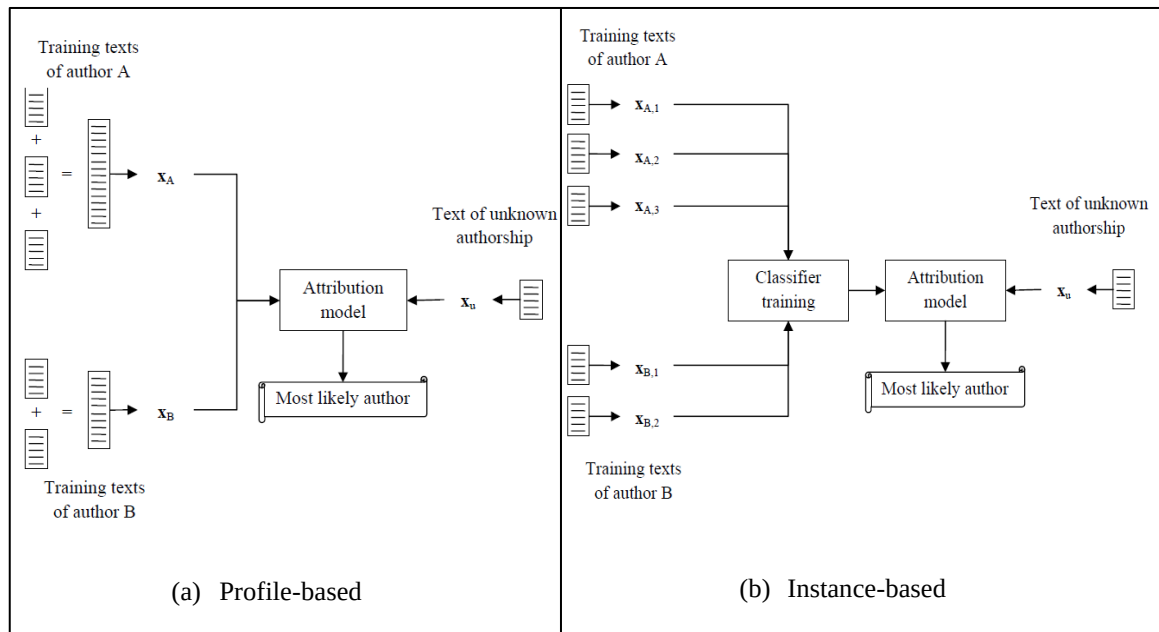


Figure 2.6: The architecture of profile-based vs. instance-based approach

Figure 2.6 depicts the difference between profile-based and instance-based approach in attributing each author's writing style. In Figure 2.6(b), the instance-based approach utilizes powerful learning algorithms that can handle high-dimensional, noisy and sparse data. Furthermore, it is easier to integrate various stylometric feature sets using an instance-based compared to a profile-based approach. As for text representation, instance-based can represent the text individually, whereas profile-based produce one cumulative representation. A study by Altakrori et al. (2018) compared the performance of AI for Arabic

tweets using profile-based and instance-based using different levels of n-grams as stylometric features. For the instance-based, they utilized four different classifiers: Naïve Bayes, SVM, Random Forest and Decision Tree. Whereas for profile-based, they adopted Common N-gram (CNG), where they used a similarity function to compare the author's profiles to the anonymous profile.

This section discusses past research that are relevant in the field of AI and SMF. The review including the type of training data used, the stylometric features that are being extracted, classifiers that are being implemented and the findings. Thus, Table 2.4 summarises the details of the attribution components for AI system from past research, specifically for short text in the SMF area.

Table 2.4: The summary of the literature review of the short text of AI task

Paper	Data	Num. of authors	Num. of samples	Attribution Method	Feature Extraction Model	Stylometric Features	Classifier(s)	Performance
Jambi et al. (2021)	Tweets (Arabic)	20	150	Instance-based	TF-IDF	• Character-based • Lexical-based • Syntactic-based	• SVM • <i>k</i>-NN • Random Forest	Character-based performed the best as individual feature. Best combination features are character and lexical-based. SVM achieved the highest accuracy with 68.5% with combination of all features.
Sarwar & Hassan (2021)	Online News Articles (Urdu)	90	985	Instance-based	Bag-of-words and TF-IDF	• Vocabulary richness • Word n-grams • Character richness • Character-grams	Probabilistic <i>k</i> -NN	All four feature sets combined achieved highest accuracy (94%)
Barlas & Stamatatos (2021)	CMCC (Cross-domain controlled Dataset)	21	2k tokens	Profile-based	Pre-trained Embedding: • BERT • ELMo • GPT-2 • ULMFiT	Token	RNN	BERT and ELMo models achieve best results
Theophilo et al. (2021)	Tweets	50	2000	Instance-based	Embedding	Character 4-grams	• XGBoost	Data augmentation technique improved the result with 74.15%
Punchihewa et al. (2021)	News articles (Sinhala Language)	16	100	Profile-based	Embedding	• Word • Character	RNN	Character-level model outperformed word-level. Highest 56.06% trained with 100 articles (high volume)
Khanh and Vorobeva (2020)	Online forum post (Vietnam)	30	200	Instance-based	Bag-of-words	• Character-based • Word-based • Structural • Content-specific	• NB • SVM • RF • LR	RF keeps the best accuracy in most attempts
Huang and Iwaihara (2020)	Tweets	50	200	Instance-based	Character embedding	• Words • Character n-grams	• CNN • LSTM	CNN-based models perform more stable when $n = 1$ and perform better than LSTM

Table 2.4 continued

Fourkioti et al. (2019)	Tweets	50	500	Profile-based	Bag-of-Words	• Lexical • Character 3-grams • Content words • POS 3-grams	• MNB • SVM • k-NN • RF	All features + SVM leads with 50.1 %
Doultani & Vijayalakshmi (2019)	Tweets	100	400,000	Instance-based	-	• Lexical-based • Character-based	Nearest-neighbour algorithm	Out of 22 features, term frequency seems helpful in finding the unknown author.
Chowdhury et al. (2019)	Blog posts (Bengali)	6	350	Instance-based	Word embedding + Skip-gram	Word n-grams	CNN	The best accuracy of 92% was achieved with CNN based on skip-gram word embeddings
Khatun et al. (2019)	Online e-Library (Bangla)	14	469	Instance-based	Character Embedding	Characters	CNN	the model performs better with a larger dataset and improved by up to 10% on pre-training.
Theóphilo et al. (2019)	Tweets	50	2000	Instance-based	Character embedding	Character 4-grams	CNN	Training accuracy 97.5 and validation accuracy 65%
Altakrori et al. (2018)	Tweets (Arabic)	20	125	• Instance-based • Profile-based	Bag-of-N-grams	• Lexical-based • Structural-based • Syntactic-based	• Naïve Bayes • SVM • Decision Tree • RF	RF using instance-based achieved the best result with 55%
Sari et al. (2018)	1. Legal judgements 2. Online news 3. Movie reviews	1. 3 2. 10 3. 62	1. 1342 2. 1000 3. 79550	Instance-based	Continuous n-grams	• Word n-grams • Character n-grams	Feed forward neural network	Character n-grams outperformed for both legal and movie reviews. Word n-grams performed well for online news.
Zhang et al. (2018)	1. Online news 2. Movie Reviews 3. Blogs	1. 10 2. 62 3. 10	1. 100 2. 62000 3. 24000	Instance-based	Word2Vec	• Syntax parse tree	CNN	Outperformed other prior art. Best result with 96.16% acc on movie dataset followed by news and blogs dataset.

From Table 2.4, lexical and syntactical features, i.e. word and character n-grams, have been proven efficient for short text. N-grams approaches are not only being represented in the BoW model but also in the embedding models and proven enhanced the accuracy of the AI system for SMF. Embedding model can capture the semantic meaning of the text in the writing style which BoW model cannot. For the attribution methods, either the instance-based or profile-based has its own advantages in handling the text representation. However, most state-of-the-art of AI for SMF relies on machine learning classification techniques that implement the instance-based approach as this approach allows a different combination of stylometric features.

2.3.4 Evaluation in Author Identification for Social Media Forensics

In machine learning, evaluating a model is crucial to building a useful machine learning model. Different evaluation metrics are used for evaluating the quality of a model's predictions. In this section, we will discuss the evaluation metrics used by previous studies in evaluating the performance of their respected AI system for SMF.

There are various ways to evaluate the performance of the machine learning model, and one of them is the confusion matrix. The confusion matrix is a contingency table, with two dimensions ("actual" and "predicted"), and identical sets of classes in both dimensions. Confusion matrix consists of rows and columns that report the number of false positives (FP), false negatives (FN), true positives (TP), true negatives (TN) as illustrated in Figure 2.7.

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Figure 2.7: Confusion matrix

Based on Figure 2.7, which in this case, the example of confusion matrix is for the binary classification. The terms "positive" and "negative" refer to the classifier's prediction, and the terms "true" and "false" refer to whether that prediction corresponds to the external judgement or observation. The definition of each term in the matrix is as follows:

- TN: Predicted negative and are actually negative
- TP: Predicted positives and are actually positive
- FN: Predicted negative but are actually positive
- FP: Predicted positive but are actually negative

From the confusion matrix, accuracy, and performance metrics, i.e., precision, recall, and F-measure, can be computed to evaluate the performance of the classification model. These evaluation metrics are as follows:

1. Accuracy: The overall rate of correctly classified examples to the total number of examples in the testing dataset and is calculated as per Eq. (1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{Equation 2.1}$$

2. Precision: The fraction of predicted positives that are actually positives and is calculated as per Eq. (2).

$$Precision = \frac{TP}{TP+FP} \quad \text{Equation 2.2}$$

3. Recall: The fraction of positive events that are predicted correctly. Is equivalent to TP-rate (Sensitivity) and is calculated as follow per Eq. (3).

$$Recall = \frac{TP}{TP+FN} \quad \text{Equation 2.3}$$

4. F-measure: Or known as F1-score is the weighted average of Precision and Recall. The higher score as a better model. It is the harmonic mean of Recall and Precision and is calculated as follow per Eq. (4).

$$F1 = \frac{2*(Precision*Recall)}{Precision+Recall} \quad \text{Equation 2.4}$$

Studies conducted by Rocha et al. (2017); Altakrori et al. (2018); Huang et al. (2020) used the same approach where they used 10-fold cross-validation to validate the performance of their system. The dataset was partitioned into training and test data sets for each experiment. The partitions were repeated ten times, and the authors included in each fold were randomly selected. The evaluation is done using accuracy. The performance of the system was analysed using a different number of authors and the number of training samples.

On the other hand, Huang et al. (2020) utilized different evaluation metrics where they used F1-score to compare the performance of their proposed models. In their study, different models were produced based on different vector size and the window size of the embeddings. They also performed a performance comparison based on the different number of training samples and authors.

Besides the cross-validation, the train/test split technique is used to validate the performance of machine learning algorithms. The splitting of data usually consists of training and test data sets. The training set usually contains labeled data, and the test sets contain unlabelled data, which will be tested to the trained model. Studies by Chowdhury, et al. (2019); Khanh and Vorobeveva (2020) used this validation technique to validate the performance of their AI systems.

However, the train/test split technique often faced overfitting and under-fitting issues, although it produced a result with higher accuracy. Overfitting happens when the model is trained "too well" and fits tightly to the training dataset. The produced model will be very accurate on training data but less accurate on new unseen data. The situation happens because the model is not generalized and unable to make any inference on new unseen data. While, under-fitting happens when the model does not fit the training data and thus, misses the trends in the data. Table 2.5 summarises previous studies on the usage of validation techniques and evaluation metrics in AI task for SMF.

Table 2.5: Previous studies based on validation techniques and evaluation metrics

Refs.	Classifier(s)	Validation technique	Evaluation Metrics
Khanh & Vorobeveva (2020)	• Naïve Bayes • Support Vector Machine • Random Forest • Linear Regression	Train/test split	Accuracy
Altakrori et al. (2018)	• Naïve Bayes • Support Vector Machine • Decision Tree • Random Forest	Cross-validation	Accuracy
Huang et al. (2020)	• Convolutional Neural Network	Cross-validation	Accuracy

Table 2.5 continued

	• LSTM		
Khatun et al. (2019)	Convolutional Neural Network	-	Accuracy
Chowdhury et al. (2019)	Convolutional Neural Network	Train/test split	Accuracy

2.4 Discussion

Stylometric measures can be used to capture information from different linguistic levels, such as morphology, lexicon, syntax, and semantics. The measures can range from simple approach, such as counting character n-grams (Punchihewa et al., 2021) or function words (Jambi et al., 2021), while others require more complex natural language processing (NLP) tools, such as part-of-speech (POS) taggers and parsers (Sari et al., 2018). Simple measures have the advantage of being language-independent, while complex measures are often more effective.

For instance, lexical features i.e. function words or unique words may not be very effective for OSN texts because OSN text contain less text to analyze which make it noisy and ineffective to identify the writing patterns that are unique enough for that particular author (Fourkioti et al., 2019). Meanwhile, n-grams are a good choice for AI in short OSN text because they are language independent and can capture specific character or word combinations (Sarwar & Hassan, 2021). The feature set can be homogeneous (Punchihewa et al. 2021), meaning that it is based on a single linguistic level, or heterogeneous (Sari et al., 2018; Fourkioti et al., 2019), meaning that it combines multiple linguistic levels. The later are usually more effective, but they can be more difficult to obtain. Several prior research

have been summarized in regards the advantages and disadvantages of the proposed solutions in the Table 2.6 below.

Table 2.6: The advantages and disadvantages of proposed solutions of prior works in regards to the stylometric features used

Paper	Proposed Solution	Advantage(s)	Disadvantage(s)
Jambi et al. (2021)	Proposed an approach for AA in Arabic short-microblog text using lexical, syntactic and character features.	Feature combination provide effective solution in capturing the distinct writing styles in this specific context focussing on the Arabic characteristic i.e. the morpheme and grammar.	The language-specific features and patterns identified may not generalized well to different linguistic context
Sarwar & Hassan (2021)	"UrduAI" for AI in Urdu text to provide a comprehensive representation of the authors based on Urdu language writing style using writeprints i.e. char n-grams, function words, punctuation, and structural features.	The use of a combination of lexical, character, and syntactic features successfully captures the unique writing style of Urdu	The proposed solution uses a number of features that are specific to Urdu i.e. the use of Urdu characters and grammatical structures but not generalised to other language
Punchihewa et al. (2021)	Proposed a language modelling approach for AI in Sinhala (Sri Lanka) text utilising n-gram models by extracting and	N-gram models allows for a comprehensive analysis of the frequency and patterns of word sequences in the	The proposed solution applicability to other domains or languages are limited where

Table 2.6 continued

	analysing the frequency of sequences of words in the text.	text which able to capture writing style and linguistic nuances specific to the authors	specific linguistic features and patterns are identified in Sinhala text
Sari et al. (2018)	Proposed a method that provide a comprehensive view of topic-based and style-based features for AI	Linguistic patterns and individual writing style such as word freq, syntactic patterns are strong indicators to distinguish authors	Topic-based features able to capture the content and subject matter but failed to perform well as this features can be influenced by the genre of the text. They unable to capture the unique linguistic patterns in the writing style.
Fourkioti et al., (2019)	Proposed a method for AI that uses fusion of language models of word-level, character-level and syntactic- level.	For tweets dataset, the combination of character n-grams and word n-grams are the best in capturing the writing style patterns while POS 3-grams facilitate as additional indicator for the effectiveness other feature set	Lexical features showed a bad performance in capturing the stylistic specialties of tweets author as the features tend to be noisy and ineffective for AI task in tweets

Previous studies relied on sparse representation such as BoW and TF-IDF (Sari & Stevenson, 2016; M. Mohsen et al., 2017; M. Mohsen et al., 2017) to represent the features. Sparse representation meaning that it only includes high-dimension of features, or dense, meaning that it includes low-dimension of features. OSN text are very noisy and

sparse in their use of vocabulary whereby sparse representations have the difficulty in grasping the synonyms or semantic connections between words, and it can be affected by homonyms (Markov et al., 2017). Pre-trained language models (i.e. Word2Vec, fastText, BERT, ELMo, ULMFiT, and GPT-2) provide dense representation with improved performance in AI task (Mahajan et al, 2018; Chowdhury et al., 2019). These models leverage neural networks and capture information from various linguistic levels by training on extensive corpora. They can be fine-tuned with domain-specific data to enhance performance.

However, pre-trained embeddings can limit model performance due to language dependence and lack of word usage consideration (Barlas & Stamatatos, 2021). Pre-trained models are trained on large corpora of existing texts, such as encyclopedias, news texts, or literature. This means that words that appear in similar contexts in training texts are represented by vectors that are close to each other. However, if a number of important words in the training text do not belong to the vocabulary of a pre-trained model, then the model will not be able to interpret these unseen words. This can result in producing “new” words without any collated embedding known as out-of-vocabulary (OOV). Such a case could arise in OSN text containing internet slang and misspelled words which are not presented in language vocabulary.

To address the OOV issue, a potential solution is to split the sentence into a list of characters instead of words. This increases the likelihood of encountering the same characters in both the training and prediction sets, improving the model's performance (Hitschler et al. 2018 proposed a model). Despite the availability of compact embedding representations for texts, the importance of n-grams remains evident in this AI task

especially towards noisy OSN text. N-grams at the word and character level have proven to be remarkably effective as standalone features. In this research, the role of n-grams and embeddings for Ai in OSN text are reinvestigated by exploring word, character, and POS n-grams as well as the use of embedding-based feature representations with these n-grams can be more effective than either approach alone. Table 2.7 depicts the advantages and disadvantages of the prior works' proposed solutions regards to the representation models.

Table 2.7: The advantages and disadvantages of proposed solutions of prior works in regards to the feature representation model

Paper	Proposed Solution	Advantage(s)	Disadvantage(s)
Markov et al. (2017)	Use character n-grams as features to represent the writing style of the author in Portuguese text.	BoW is used to represent the n-grams due to its simple, computational inexpensive and language-independence	The BoW approach, with low features included (which related to topic information) makes the result biased since the model captures topic information rather the stylistic information. The use of low freq words in TF-IDF will deteriorate performance, adding noise and creating sparse representation.
M. Mohsen et al. (2017)	Proposed method for deep learning authorship attribution using variable size character n-grams	Bag-of-words of frequent words including stopping words are easy to implement and straight forward	Perform rather low due as the number of features increase and the features included mostly capture only topic-related info

Table 2.7 continued

Chowdhury et al. (2019)	Proposed a method for authorship attribution in Bengali literature using CNNs with fastText's word embedding model	fastText is able to capture both local and global features of the text, which makes it more robust to noise and variations in writing style.	The model was trained on a relatively small dataset of Bengali poems and using only one type of feature type. This could limit the generalizability of the model to other datasets or languages.
Barlas & Stamatatos (2021)	Proposed a transfer learning approach to cross-domain authorship attribution using pre-trained models	pre-trained models help to obtain higher performance compared to self-trained for cross-domain AI in controlled corpus	Restrictions imposed by the use of pre-trained language models i.e. input size restriction and it complicates the analysis of long documents. Unable to combine character-level and token-level information within a single model. Language dependent.

Machine learning models i.e. Naive Bayes, Random Forest and SVM struggle with high-dimensional and sparse linguistic data. On the other side, dense representations are more easily combined with powerful classification algorithms i.e. deep neural network (DNN), since they provide relatively low-dimensional feature sets. DNN has been shown to be a promising approach for authorship attribution. Different types of DNN models have been used for authorship attribution, including CNNs, RNNs, and transformer models. Sequential-based neural networks i.e. RNNs and transformer-based are good for variable-length data like text and are used for tasks like language modeling, machine translation and speech recognition. CNN excels in handling data with spatial relationships.

While CNNs typically accept two-dimensional inputs, such as matrices, they can also be adapted for one-dimensional sequences. This flexibility allows CNNs to be applied to various data types with spatial relationships. OSN texts that tend to be noisy and informal. Sequential model (i.e. LSTM, Bi-LSTM, and GRU models) excel in capturing long-range dependencies and sequential information in text, while CNN model is more suitable for capturing local patterns and handling variable-length inputs, making them a suitable choice for AI in OSN texts (Chowdhury et al. 2019). However, conventional ML model like SVMs can outperform DNNs on text classification tasks, if the parameters and features are carefully selected and there is a sufficient amount of text data with a lower number of classes (Romanov et al., 2021). Table 2.8 shows the advantages and disadvantages of the proposed solutions in regard to the classifiers used.

Table 2.8: The advantages and disadvantages of proposed solutions of prior works in regards to the classifier used

Paper	Proposed Solution	Advantage(s)	Disadvantage(s)
Chowdhury et al., (2019)	The proposed solution involves training CNN models on fastText word embeddings to identify the authors of Bengali texts.	CNN with word embeddings as features outperformed RNN with LSTM as it able to capture the context and style of the different authors from the sequence of words very well in this scenario.	CNN can be less effective than RNNs at capturing long-range dependencies in text. DNN model require fine-tuning based on the text sample.
Barlas & Stamatatos (2021)	Proposed a transfer learning approach to cross-domain authorship attribution using a pre-trained	Used a variety of pre-trained language models (BERT, GPT-2, ELMo and ULMFiT) which help to improve the accuracy compared to self-trained	All of the pre-trained LMs need to fine-tuned for the specific AA task.

Table 2.8 continued

	language model to represent the text of the documents	RNN. BERT and ELMo achieved the best results for cross-domain while UMLFiT and GPT-2 are not that competitive.	There are differences in the parameter settings of each model to obtain as high performance as possible
Romanov et al. (2021)	Proposed a method for AI of Russian-language texts using SVM and deep neural networks (DNNs) i.e., LSTM, CNN with Transformer	SVM precede the DNN models with due to thoroughly chosen parameters and feature space, which includes statistical and semantic features.	DNNs are inferior to SVM in accuracy as DNNs require more data for learning. SVM classification is based on an accurately found set of features hand-crafted by experts.

According to Kešelj et al. (2003) the most common approach to determining authorship is to use stylistic analysis that proceeds in two steps: first, specific style markers are extracted, and second, a classification procedure is applied to the resulting description. Hence, this research makes three key contributions. Firstly, by utilizing only standard tags, native information from the OSN platform were efficiently embedded with the stylometric information, resulting in improved outcomes.

Secondly, we investigate various levels of n-grams and their representations, including sparse and dense approaches. For dense representation, we employed pre-trained model for English text to enhance the performance and as for under-resourced language (U-RL) text, self-trained model is produced to avoid OOV issue. Furthermore, we independently feed the representation models into diverse classifiers, including both machine learning and

deep learning models. This comprehensive analysis provides insight to enhance the performance of the proposed technique of AI for OSN text analysis.

2.5 Summary

The rise of OSN has led to an increase in cyberbullying. Cyberbullying is difficult to track down because perpetrators can remain anonymous. Digital forensics can be used to capture digital evidence from OSNs, but this does not reveal the identity of the perpetrator.

AI can be used to identify the author of an anonymous message. This can help investigators to narrow down the pool of suspects. However, AI task is challenging in OSNs because of the short text length and the continuous evolution of internet language. Additionally, OSN often contains cross-domain content, which can further complicate the task of AI.

Previous work on AI n OSN text has shown that dense representation models combined with deep learning algorithms can be effective for short text tasks but conventional ML approach plays a crucial role in feature extraction and selection. Thus, this research explore different combinations of feature set, representation models and classifiers and make analysis on each model to produce an AI system that could maintain its performance in analysing OSN text. The mind map for Chapter 2 is included in Appendix C.

CHAPTER 3

METHODOLOGY

The background study of digital forensics and Author Identification (AI) in mitigating Internet issues, for instance, cyberbullying in social media has been discussed in Chapter 2. Author Identification (AI) can facilitate digital forensics to reveal the most plausible author of an anonymous message by analysing the candidate authors' writing styles. However, the challenge of state-of-the-art AI in Social Media Forensic (SMF) lies in the limited information of short text and Internet parlance in the social media text. Hence, proper mechanisms of AI system are needed especially for SMF to handle social media text in producing optimal accuracy and improving AI performance.

This chapter introduces the proposed workflow of AI in SMF. In general, the workflow of AI adopts text classification process but distinct in a way in regards to the analysis of the writing style based on the candidate authors (Howedi & Mohd, 2014). Figure 3.1 illustrates the general workflow of text classification.

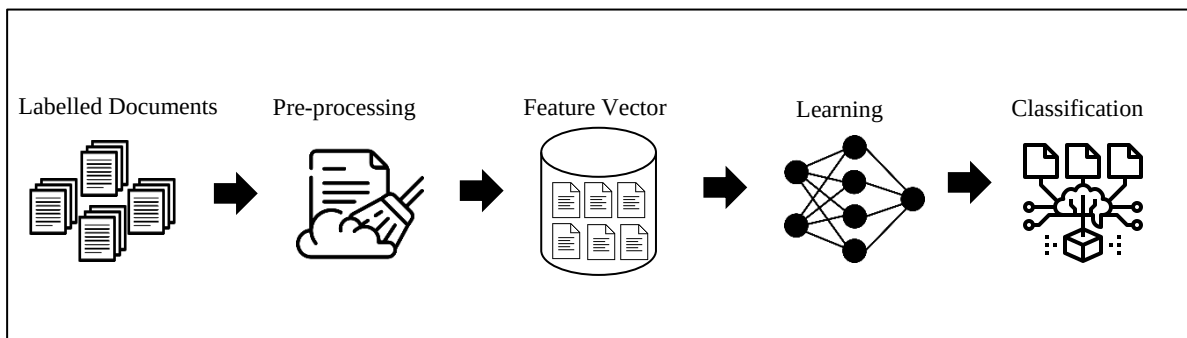


Figure 3.1: General text classification workflow (Ferreira, 2017)

3.1 The Proposed Workflow of Author Identification for Social Media Forensics

Figure 3.2 shows the overall flow of the proposed workflow of AI for SMF. The proposed workflow is divided into three phases: data preparation, classification phase and evaluation phase whereby each phase is customized according to the research contribution.

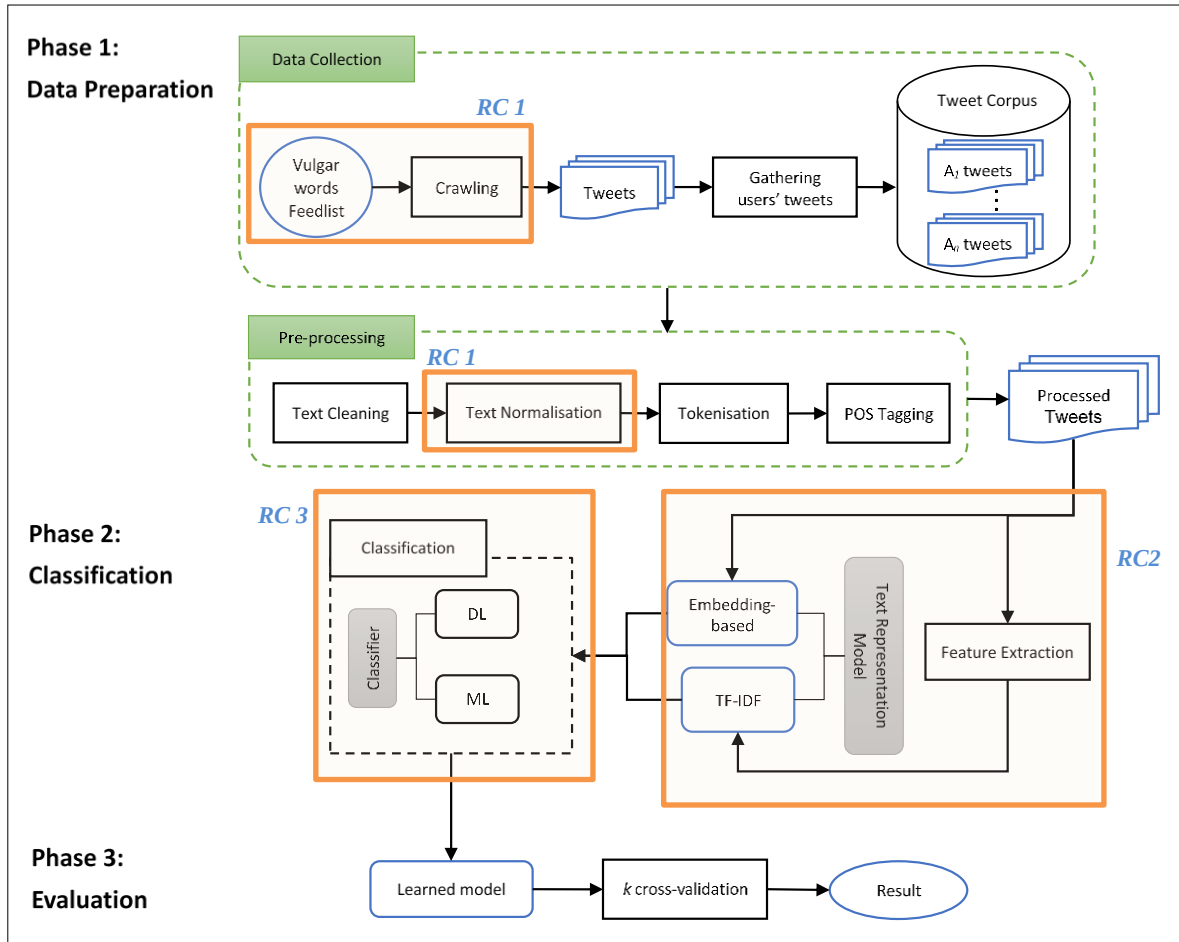


Figure 3.2: The proposed workflow of AI for SMF

Section 3.2 explains the preparation of data where the tweets are crawled and undergone pre-processing. Section 3.4 explains Phase 2, where the formation of text representation models and classification of text taken part. The machine learning (ML) and deep learning (DL) approach which are used to classify the text based on the authors are

discussed. Section 3.5 explains the evaluation phase discussed the evaluation matrices used to measure the performance of the learned models also, the validation technique used for the validation process.

3.2 Text Preparation Phase

The preparation of the data set is used to test the proposed workflow is explained in this section. The description includes the type of text that is used to be analysed in AI for SMF. The following sub-sub sections explain the data collection and pre-processing involves crawling the sample text from popular OSN platform, Twitter, and cleaning the text to turn them into standard text.

3.2.1 Collecting Tweets from Twitter

The proposed workflow of AI in SMF aims to identify the author of an anonymous online social network (OSN) message from a set of candidate authors based on OSN text. Tweets from Twitter are used as the source of data in representing the short text in OSN. Twitter is a popular OSN platform that allows data to be acquired easily using Twitter API. Twitter is a microblogging service type OSN that enables its users to publish short messages called tweets containing 280 characters (or less) including text, video, GIFs, and web links.

A microblog is a short content with the purpose of quick audience interactions. Microblogging combines the idea of instant messaging and content production, where short messages are shared online to improve engagement with the audience. Twitter users have their unique username, which starts with a "@" sign followed by alphanumeric characters without space, e.g., @Username. Twitter also allows grouping of tweets or topics using the hashtag "#" for instance, #machinelearning.

Moreover, the sharing of images, videos, and URL links are permitted in tweets. The usage of emoji and emoticon is supported too. Emoji is a collection of pictographs used to express facial emotion or ideas represented as an encoded character, e.g., 😊, 🙏. Meanwhile, emoticons are combinations of punctuation marks, letters, and numbers, to create pictorial icons, e.g., "=D." Tweets hold unique and special features that could be used as additional information to attribute a message to its author.

Data collection (ref. Figure 3.2) consists of two sub-processes: First, a list of recent tweets is crawled using feedlist. After that, additional tweets from different users are gathered using the unique *User_ID* which are obtained from the crawled tweets before. Through the *User_ID*, tweets from a specific user are acquired to learn the writing style of each author later.

a. Crawling Tweets using a Feedlist of Vulgar Words

The restrictive terms set by most OSN platforms, including Twitter, prohibited the dissemination of such data set. As there is no public data set for AI in SMF thus, tweets are crawled using feedlist to build a data set of authors with short OSN text. As illustrated in Figure 3.3, the data preparation phase starts with the crawling process.

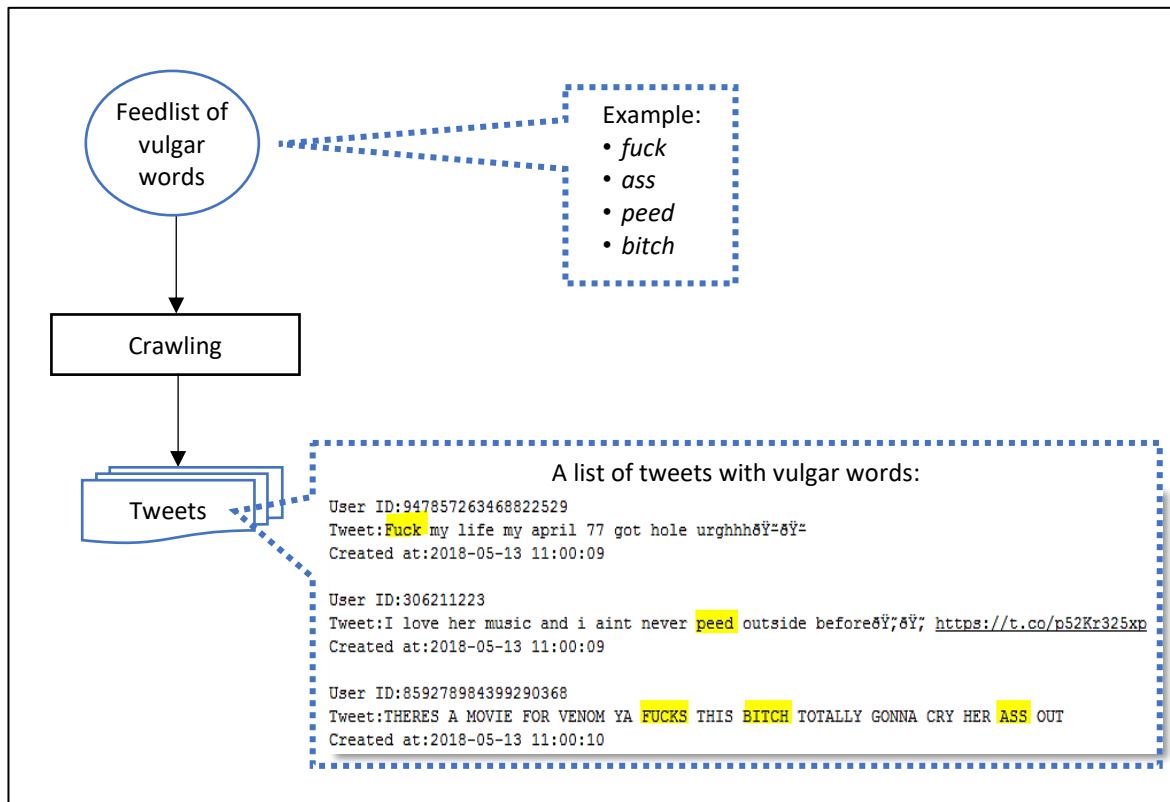


Figure 3.3: Crawling tweets using keywords in the feedlist

In Figure 3.3, the feedlist consists of a list of vulgar words obtained from a website, non-swearing.com. The website provides a list of common English vulgar words that are often used by Internet users. The list has 370 of cursing words. The feedlist is inserted into a crawler, Tweepy, a Python library used to access Twitter API to crawl tweets. Tweepy tracked down the keywords in the feedlist and returned a list of recent tweets which contained the keywords. There were 50 recent tweets randomly streamed via Twitter API class, *TwitterStreamListener*, a listener that handles tweets received from the stream. The listener returns a list of 50 recent tweets as of 13th May 2019 containing profane words from the feedlist. The streaming method allows downloading random tweets in real-time.

b. Gathering Tweets from Each Twitter User

After crawling, the *User_ID* obtained from the crawled tweets are used to acquire tweets from each user. The Twitter user IDs are unique thus assumed that each user is a different person. Based on the proposed workflow, the tweets act separately as a unit to represent the author's writing style which is called an instance-based approach. Figure 3.4 illustrates how a user's tweets are extracted from user's timeline using Twitter API "*get_all_tweets*" method.

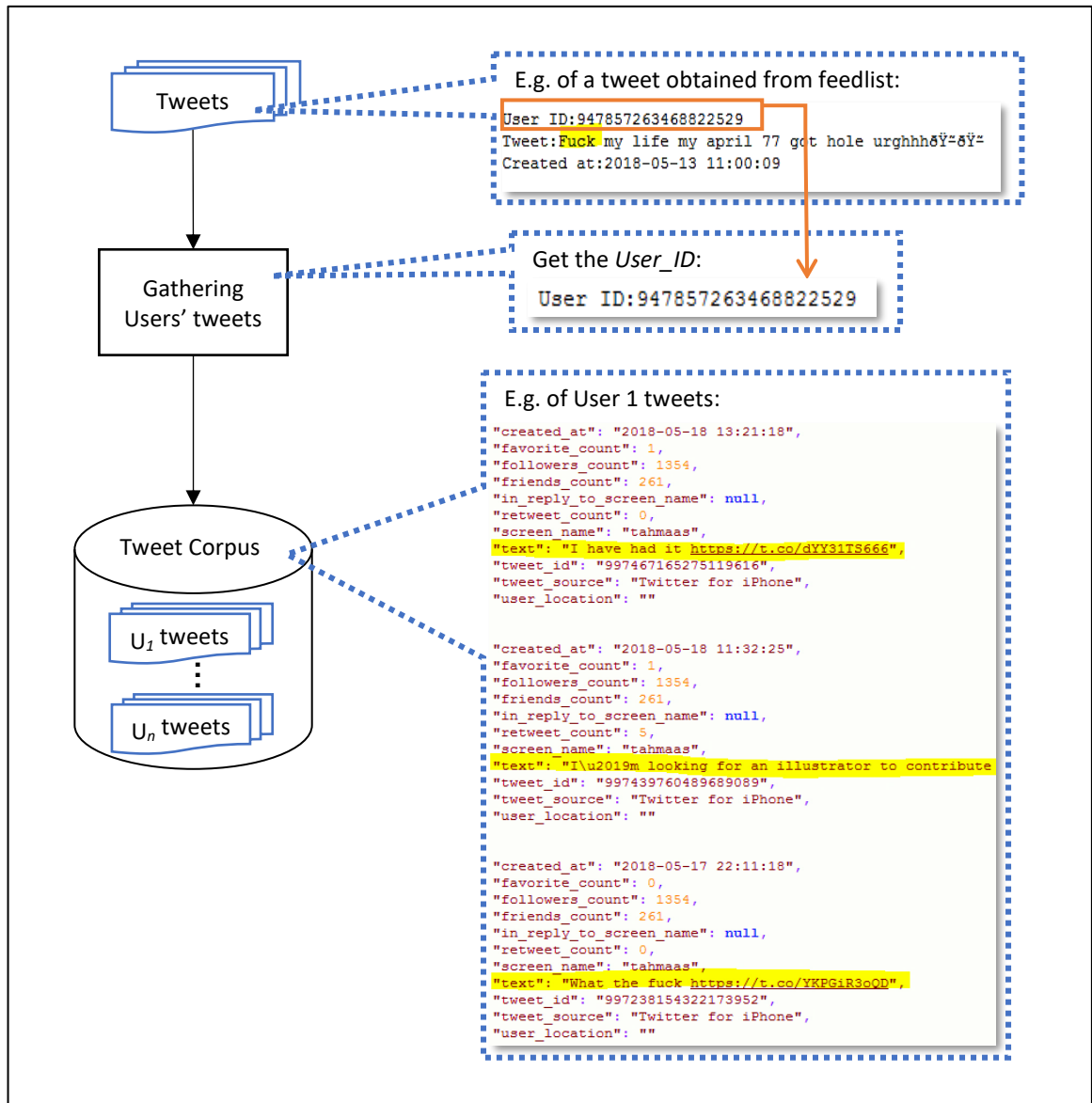


Figure 3.4: Process of gathering tweets from Twitter users

The number of tweets of the users varies based on their posting activities. Appendix B presents the summarisation of the collected tweets. It shows the number of tweets of 50 Twitter users that have been crawled through the users' timeline. There are differences in the number of tweets gathered from the users. Figure 3.5 depicts the power law distribution graph of the number of tweets gathered from each user.

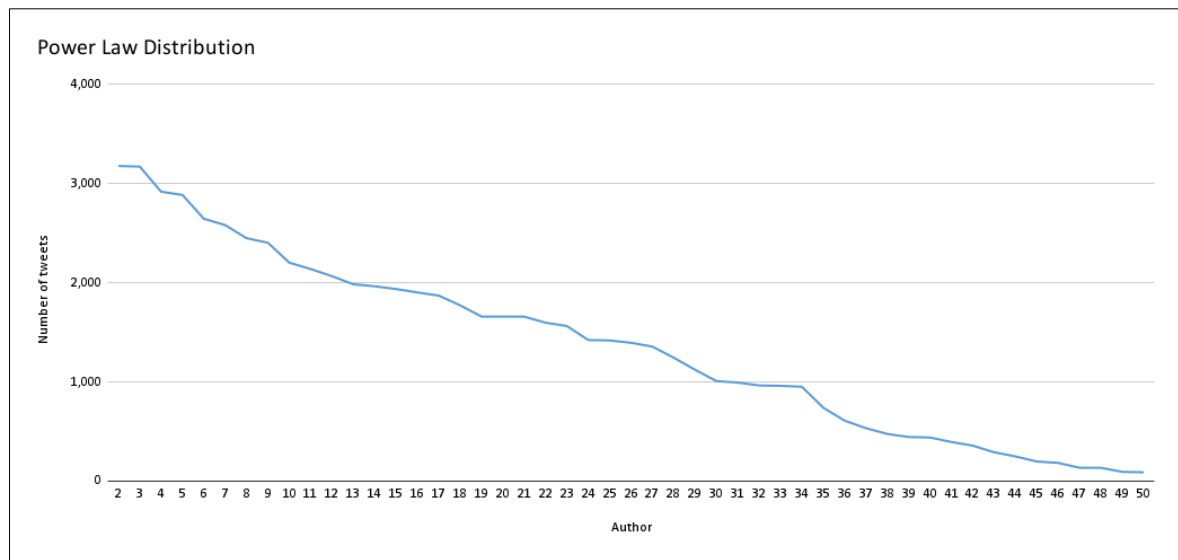


Figure 3.5: Twitter users with their respective number of tweets

According to Figure 3.5, the highest number of tweets collected is 31, 188, and the lowest number of tweets collected is 89. Figure 3.6 illustrates the total number of tokens against the number of vulgar word occurrences for each user.

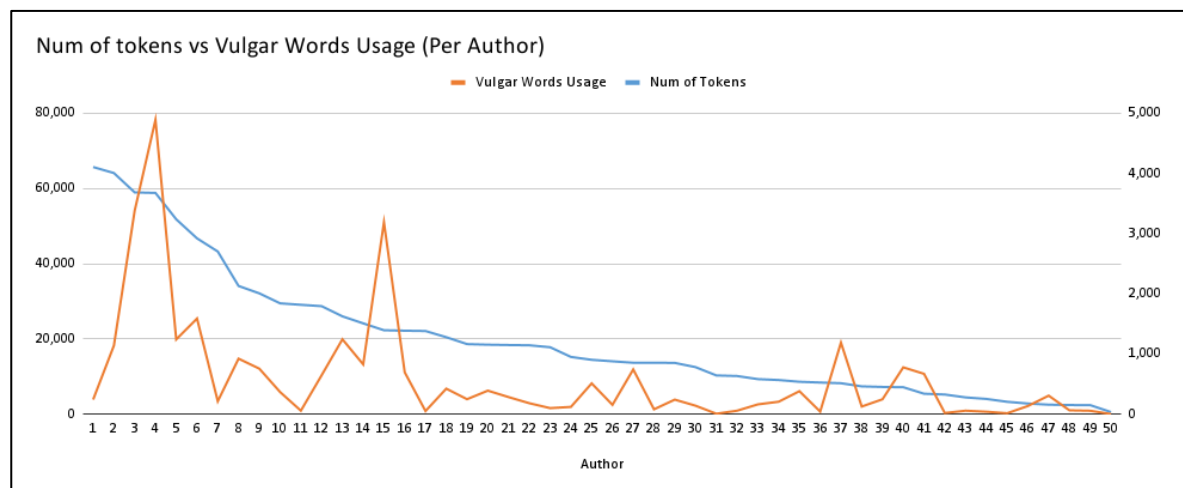


Figure 3.6: Total number of tokens and vulgar word occurrences of each user

Figure 3.6 illustrates User 1 has the highest number of tweets with 3,188 tweets compared to User 8, with 2,450 tweets. User 1 has fewer tokens with 22, 316 tokens than User 8 with 65, 871 tokens. Yet, User 8 has a low total number of vulgar word occurrences with only 244 words compared to User 1, with 3,187 total vulgar words occurrences.

The inference from the graph in Figure 3.6 explains that a high number of tweets and tokens do not define that a user has a high number of vulgar words occurrences. However, users with high vulgar word occurrences in their tweets have a higher probability of conducting cyberbully activities on Twitter. Therefore, high usage of vulgar words in OSN text posted by a specific user is especially prone to cyberbullying.

After gathering the tweets for each user, the tweets will undergo text pre-processing for text cleaning, and the tweets are transformed into standard text. The next subsection explains the pre-processing step in detail.

3.2.2 Pre-processing of Tweets to Standard Text

The tweets gathered are in JSON format, which contained text with full metadata provided by the Twitter API. The pre-processing includes: the first step is text cleaning; the second step is the text normalisation; the third step involves tokenisation; the last step is POS tagging. Figure 3.7 depicts the steps in pre-processing.

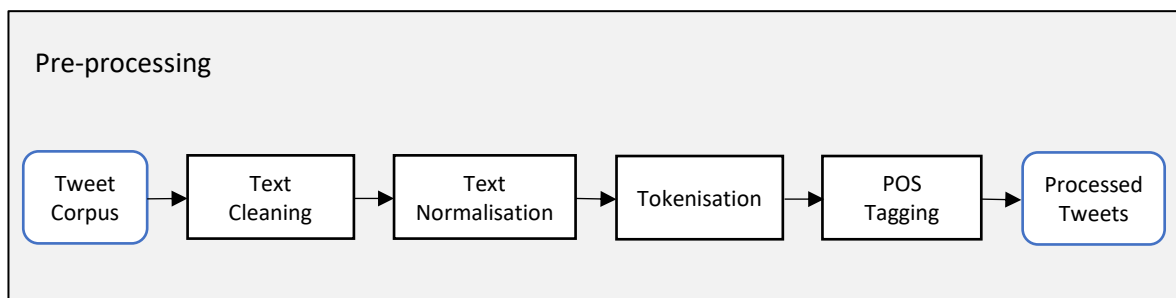


Figure 3.7: Pre-processing of collected tweets

a. Text Cleaning

Before the cleaning process is done, all tweets in the JSON file are converted into CSV format. Figure 3.8 depicts the conversion and cleaning process of collected tweets. The cleaning process is done in two phases, first by cleaning the metadata and then through normalisation.

During the conversion from JSON to CSV, retweets are filtered out with a specific retweet flag and the meta tag 'RT.' Since retweets are not the original writing from the user, they should be filtered out. Tweets with a meme (an image macro paired with a concept or catchphrase) are filtered out too. Tweets with meme contain only images and catchphrases that do not represent the author's writing style.

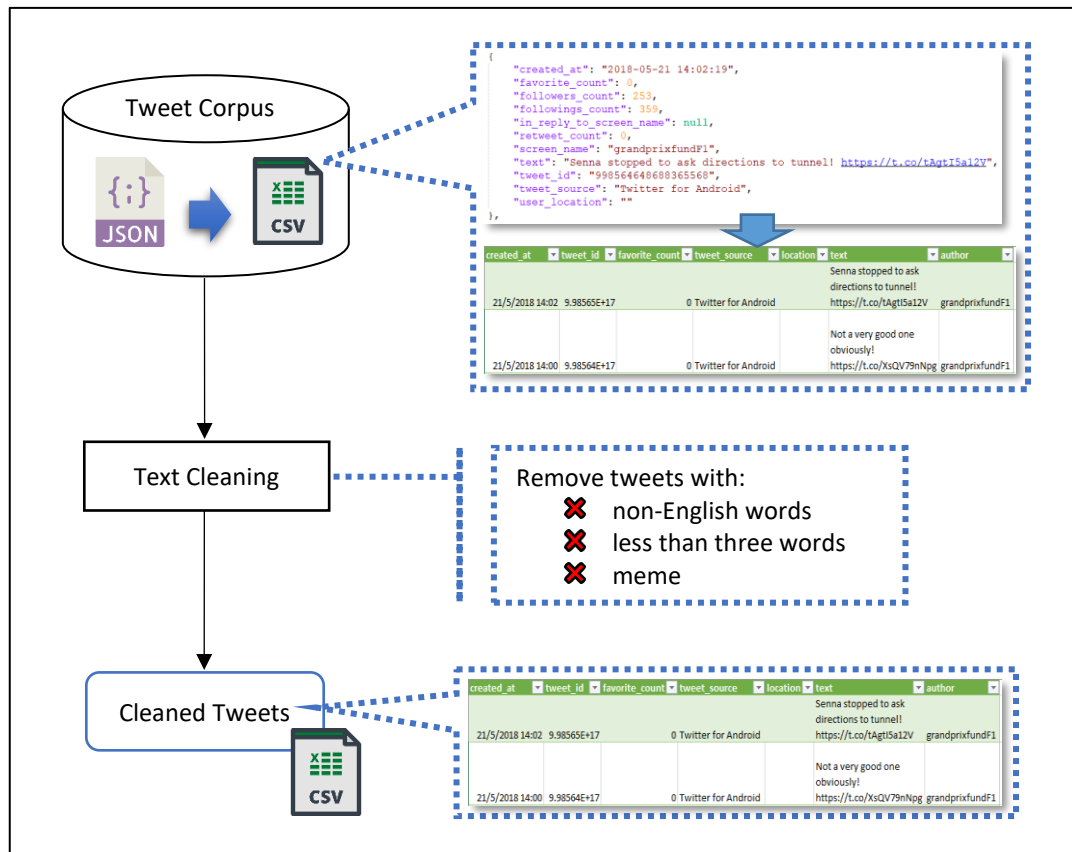


Figure 3.8: Cleaning the tweets from undesired data

As shown in Figure 3.8, once the files are converted into CSV, the cleaning process continued by removing non-English tweets, as this thesis focus on English language tweets only. Tweets with less than three words are removed because these too short messages do not provide meaningful information about the author's writing style and end up introducing noise into the classification task.

b. Normalisation

After cleaning, text normalisation is done to normalise the tweets into standard text. Normalisation is an essential process in retaining salient information like Twitters' native features. Keeping the native features will expand the feature dimension during the feature extraction process. However, normalisation reduces the number of dimensions by replacing

the original sparse characters to characters with standard tags without eradicating the information.

Figure 3.9 illustrates the normalisation of tweets where sparse characters such as numbers, dates, and times in the tweets are normalised into standard tags. Native features of Twitter, i.e., hashtags, user references, and URLs, are also being replaced with standard tags.

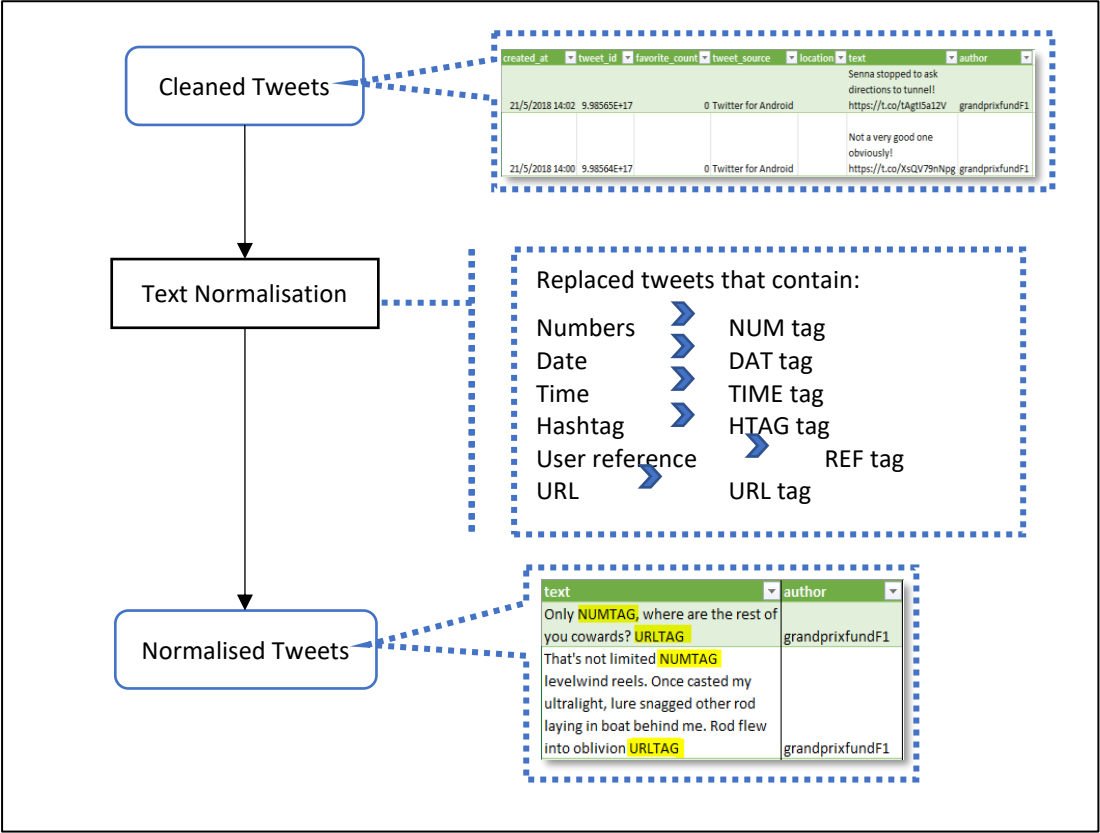


Figure 3.9: Normalising the tweets into standard text

In Figure 3.9, normalisation returned new CSV files with only two columns; the first column consists of the tweets, and the second column is the author column consisting of the user screen name. The pre-processing of the texts will continue to text tokenisation and POS tagging. Table 3.1 shows a comparison of tweets before and after the normalisation process.

Table 3.1: Example of tweets before and after normalisation

Before normalisation	After normalisation
"Can't wait this coming 9/5/2018"	"Can't wait this coming DAT"
"Help @jojoyohenes to go back to his hometown #GE14 #carpool. Plis click the link ^^ https://www.pulangmengundi.com"	"Help REF to go back to his hometown HTAG HTAG. Plis click the link ^^ URL"

c. Tokenisation

The process of tokenisation is achieved by using NLTK (Natural Language Toolkit) tokeniser module, *TweetTokenizer*. Figure 3.10 shows the tokenisation process being done.

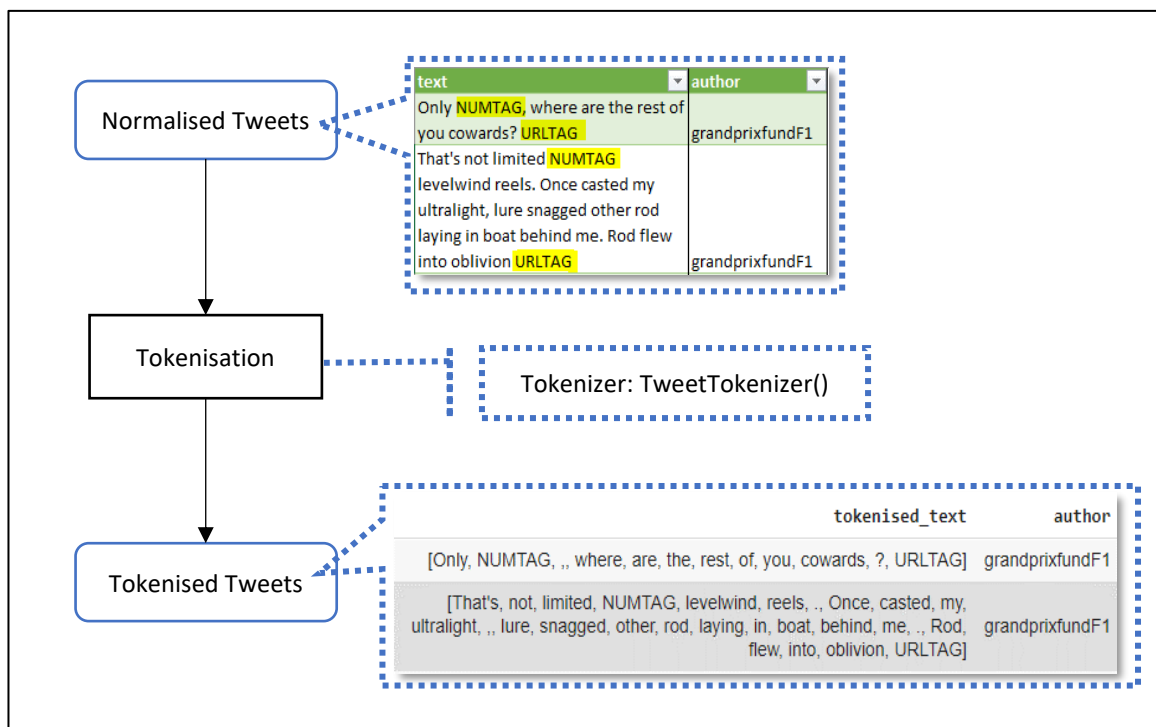


Figure 3.10: Tokenisation of tweets

Based on Figure 3.10, *TweetTokenizer* is deployed to tokenise the tweets because it is more adaptable to the tweet structure than the standard tokeniser. Table 3.2 shows the comparison of *TweetTokenizer* and standard tokeniser in tokenising a tweet.

Table 3.2: Tokenisation of a tweet using *TweetTokenizer* and *word_tokenize*

Tweet	TweetTokenizer	word_tokenize
This is a coool #dummysmile: :) :-P <3 and some arrows < > -> <--	'This', 'is', 'a', 'coool', '#dummysmile', ':', ':)', ': -P', '<3', 'and', 'some', 'arrows', '<', '>', '->', '<--'	'This', 'is', 'a', 'coool', '#', 'dummysmile', ':', ':', '-', ')', 'arrows', '<', '>', '-', '>', '<', '--'

d. Part-of-Speech Tagging

After tokenisation, continue with POS tagging. In POS tagging, the tokenised tweets are tagged or labelled with their corresponding part-of-speech grammar. POS tagging is vital to avoid noise in stylistic measures that can hinder the AI model's performance for short text. POS tags also can reduce the feature-space to a limited number of very general elements. An off-the-shelf tagger, *pos_tag*, is used to tag the tokens based on the *Penn Treebank* tagset. Figure 3.11 shows POS tagging on the tokenised tweets using the off-the-shelf tagger.

After being POS tagged, the tweets are in the form of standard text along with tagged tokens. Each tweet represents the author which is the Twitter user. These tweets will go through the second phase of the proposed workflow, which is the classification phase. In the second phase, the text is transformed into numerical vector representations for classification.

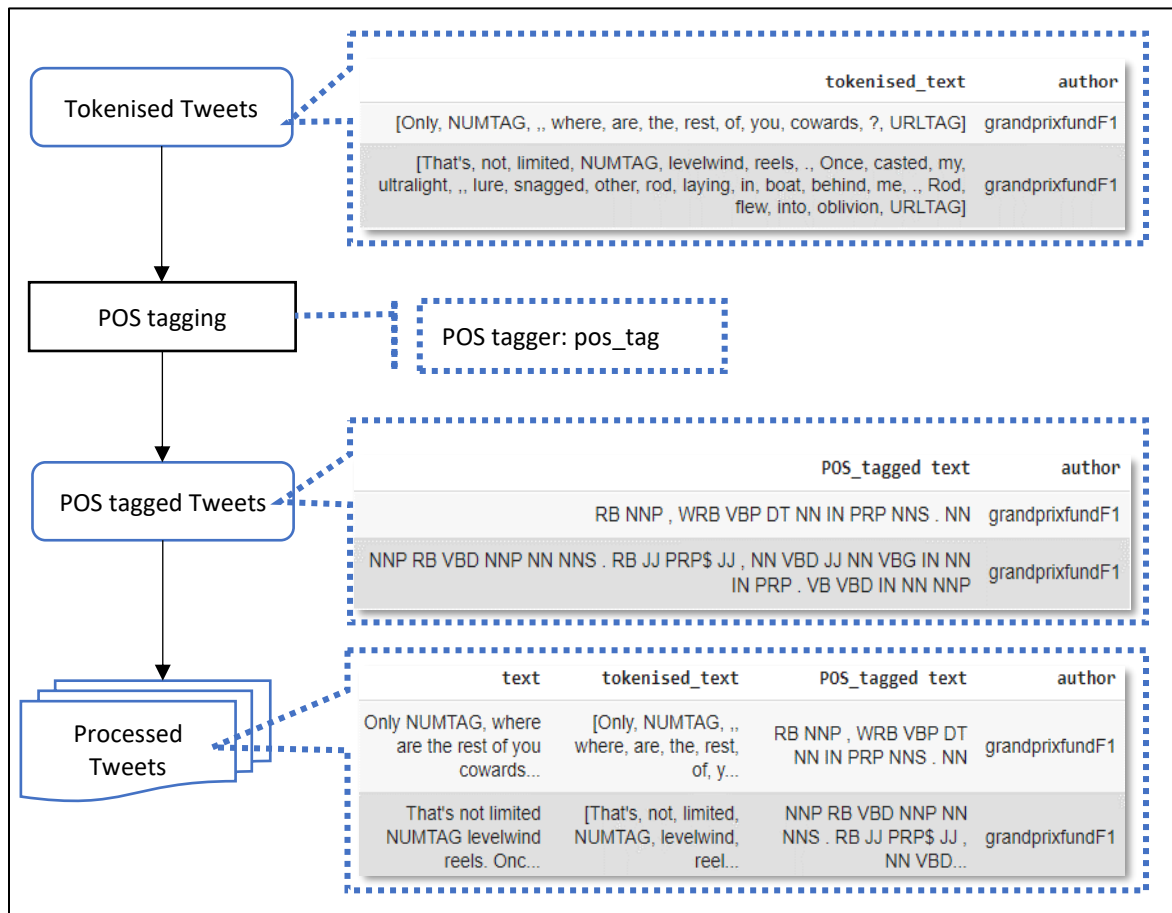


Figure 3.11: POS tagging on tokenised tweets

3.3 Text Classification Phase

Tweets are now in the form of standard text with POS information. All tweets with their respective authors will enter the second phase of the workflow, the classification phase. In this phase, the processed tweets are transformed into a vector of matrices. After that, the vectors are passed to the classification algorithm to be trained and produce learned AI models. The second phase of the workflow is pictured, as in Figure 3.12.

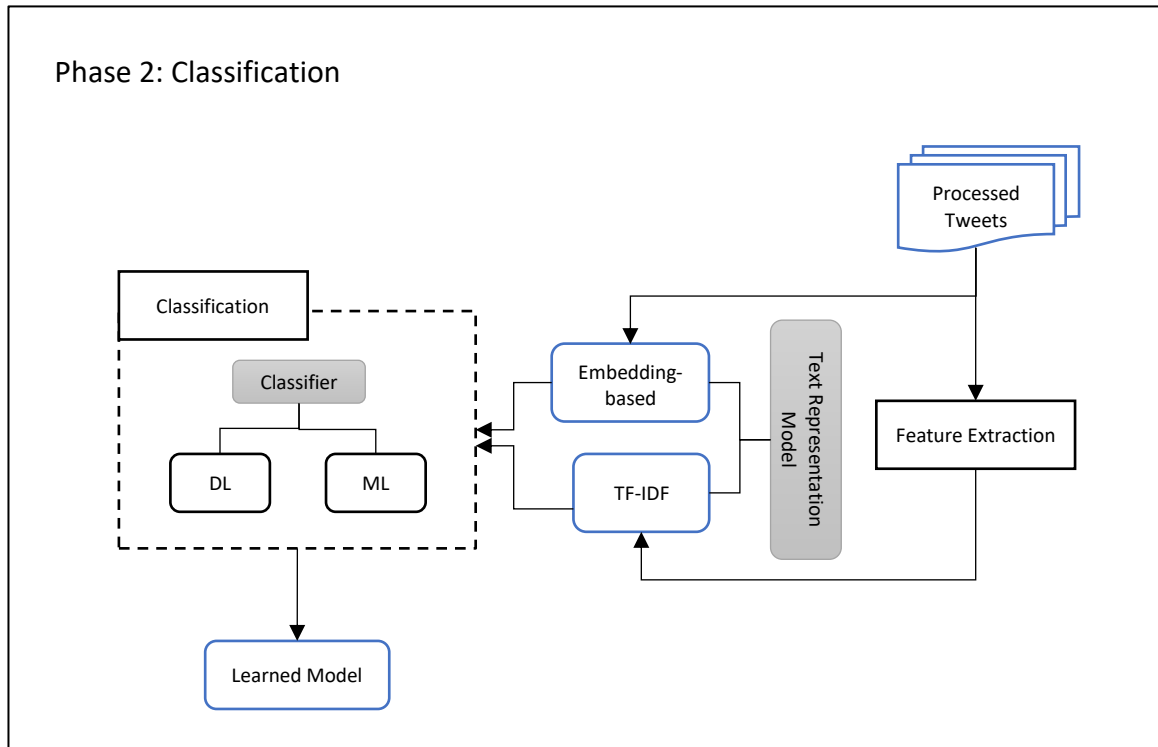


Figure 3.12: Text classification phase

Figure 3.12 shows that after the features being extracted, the features are represented in vector representations in numerical value to be learned by the classification algorithms. There are two types of text representation models being deployed to represent the features, TF-IDF and embedding-based representation. After that, classification algorithms learn the vector representations by classifying the tweets and building a learned AI model.

3.3.1 Building Text Representation Model for Tweets

Based on the proposed workflow, processed tweets are represented using TF-IDF model extracting N-grams as features. N-grams are chosen as features because N-grams work well in capturing heavy usage of punctuation, high spelling error, and Internet parlance usage in OSN text like tweets. Besides TF-IDF, embedding-based representation is

employed as the alternatives to overcome sparsity issue in TF-IDF N-grams model on short text.

a. TF-IDF Representations

TF-IDF is used to weight the N-grams of tweets. TF-IDF evaluates how important each N-gram is to a tweet in the whole collection of tweets. The importance increases proportionally to the number of times the N-gram appears in the tweet but, is offset by the N-gram frequency in the corpus.

TF-IDF weight first computes the normalised Term Frequency (TF), which appears to be the number of times an N-gram appears in a tweet, divided by the total number of N-grams in that tweet.

$$TF(n) = \frac{\text{(Number of times ngram } n \text{ appears in a tweet)}}{\text{(Total number of ngrams in the tweet)}} \quad \text{Equation 3.1}$$

The Inverse Document Frequency (IDF) is then computed as the logarithm of the number of tweets in the corpus, divided by the number of tweets where the specific N-gram appears.

$$IDF(n) = \log\left(\frac{\text{Total number of tweets}}{\text{Number of tweets with ngram } (n) \text{ in it}}\right) \quad \text{Equation 3.2}$$

N-grams have been proved to successfully capture detailed stylistic information on the author's lexical, syntactic, and structural preferences. N-grams also produced outstanding results in classifying authors of short text and multilingual text as they are capable of

indicating grammatical and orthographic tendencies without the need for linguistic background knowledge. Figure 3.13 illustrates the process of extracting N-grams features from a sequence of words in tweets into numerical value in TF-IDF model.

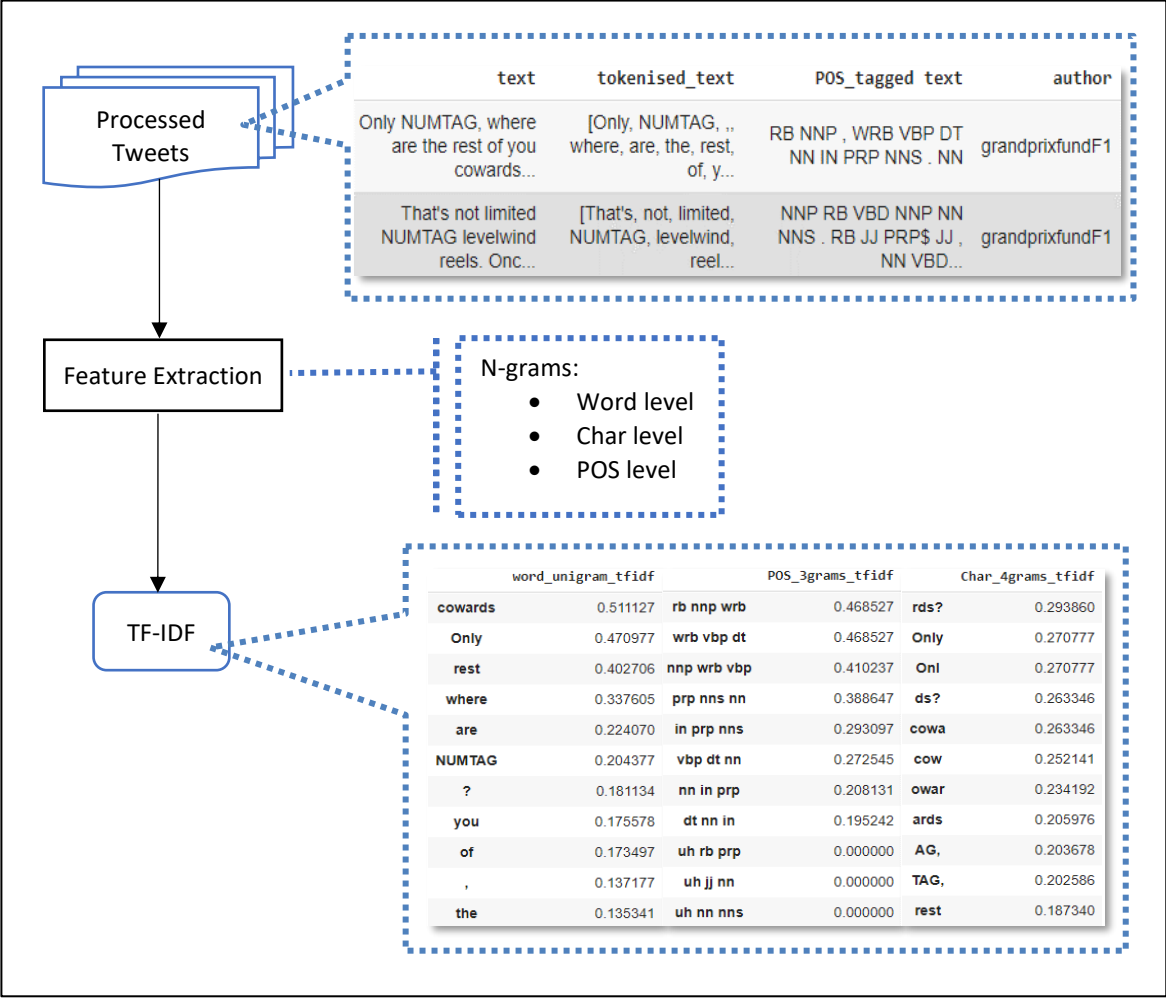


Figure 3.13: Feature extraction of N-grams in TF-IDF model

In Figure 3.13, the N-grams are extracted as features to represent the tweets in the TF-IDF model. TF-IDF representations of N-grams are explored at the word, POS (Part-of-Speech), character-level, and the mixture of all three to extract the authors' stylometric features the tweets.

i. Word N-grams

Word n-grams consist of groups of one, two, or more sequences of words capable of capturing more semantically meaningful information from a text in the form of a sequence of words. The limited length of text in tweets curb the users to be more sensible in their writing. Thus, it is logical to assume that the authors will economise and only choose short phrases. Punctuation sequences also are included as they might be a significant component of the phrase.

Word unigrams, bigrams, 3-grams, 4-grams, 5grams, and (1-5)-grams are explored to analyse their impact in distinguishing the author's writing style. All word level n-grams are case-sensitive. The following example shows a pre-processed tweet and its corresponding unigrams, bigrams, 3-grams, and (1-3)-grams.

**for readability, ' _ ' represents whitespace)*

Tweet	: “Can’t wait this coming <u>DAT</u> ”
Tokenised	: [“Can”, “”, “t”, “wait”, “this”, “coming”, “DAT”]
Unigrams	: [“Can”, “”, “t”, “wait”, “this”, “coming”, “DAT”]
Bigrams	: [“Can_”, “_t”, “t_wait”, “wait_this”, “this_coming”, “coming_DAT”]
Word 3-grams	: [“Can_ _t”, “_t_wait”, “t_wait_this”, “wait_this_coming”, “this_coming_DAT”]
Word (1-3)-grams	: [“Can”, “”, “t”, “wait”, “this”, “coming”, “DAT”, “Can_”, “_t”, “t_wait”, “wait_this”, “this_coming”, “coming_DAT”, “Can_ _t”, “_t_wait”, “t_wait_this”, “wait_this_coming”, “this_coming_DAT”]

ii. Character N-grams

Apart from being language-independent features, character N-grams need minimal text processing yet, proven to be very useful in AI. They can capture unusual features in OSN text, such as emoticons and unique use of punctuation. They can also capture the

nuances of style, the hint of contextual information, and tolerant noise. Character 3-grams and 4-grams are explored, and both features are compared in improving the performance of AI for SMF. The following example shows a tweet from the previous example with its character 3-grams features.

**for readability, ‘_’ represents whitespace*

Tweet : “Can’t wait this coming DAT”
Tokenised : [“Can”, “”, “t”, “wait”, “this”, “coming”, “DAT”]
Char 3-grams : [“Can”, “an”, “n’t”, “t_”, “t_w”, “_wa”, “wai”, “ait”, “it_”, “t_t”, “_th”, “thi”, “his”, “is_”, “s_c”, “_co”, “com”, “omi”, “min”, “ing”, “ng_”, “g_D”, “_DA”, “DAT”]

The character features are case-sensitive and able to capture the author’s preference in capitalising a letter. Letter capitalisation can be used as a trait to distinguish the writing style of each author. Usually, social media users tend to use capitalised words in expressing and exaggerating their writings.

iii. POS N-grams

POS N-grams are considered as simple stylistic features that are related to the syntactic structure of texts. POS N-grams have been used to represent the style of text that is usually unrestricted, and it performs quite accurately. POS N-grams efficiently handle noise in stylistic measures that can favourably affect the attribution model's performance. The following example shows the POS 3-grams for a tweet taken from the previous example above.

**for readability, ‘_’ represents whitespace)*

Tweet : “Can’t wait this coming DAT”
Tokenised : [“Can”, “”, “t”, “wait”, “this”, “coming”, “DAT”]
POS tagged : [“MD”, “VB”, “JJ”, “NN”, “DT”, “VBG”, “NNP”]
POS (1-3)-grams : [MB_VB_JJ, JJ_NN_DT, DT_VBG_NNP]

iv. Mix N-grams

The combination of word, POS, and character N-grams is explored to observe the impact of mix stylometry features in attributing the tweets to the authors. The experiments will repeat by combining different levels of word N-grams with POS and character N-grams (3-grams and 4-grams). The combinations involve concatenating features from each level of N-grams. After that, the best combination of the N-grams will be chosen based on the highest accuracy among the combinations.

However, the downside of N-grams is that they have very little sense about the semantics of words. N-grams model could not project the distances between words; e.g., the words “cat,” “meow,” and “selfish” are equally distant even though semantically, “cat” should be closer to “meow” compare to “selfish.” OSN messages like tweets often contain few words; thus, by using N-grams, especially character N-grams, will significantly increase the feature vectors' sparsity.

b. Embedding-based Representations

Recently, distributed embedding representations became popular and useful alternatives to sparse representation of text in classification (Barlas & Stamatatos, 2021).

Embedding model resulted in dense vector with a fixed size of n -dimension. Figure 3.14 illustrates the flow of the embedding models creation for tweets in the proposed workflow.

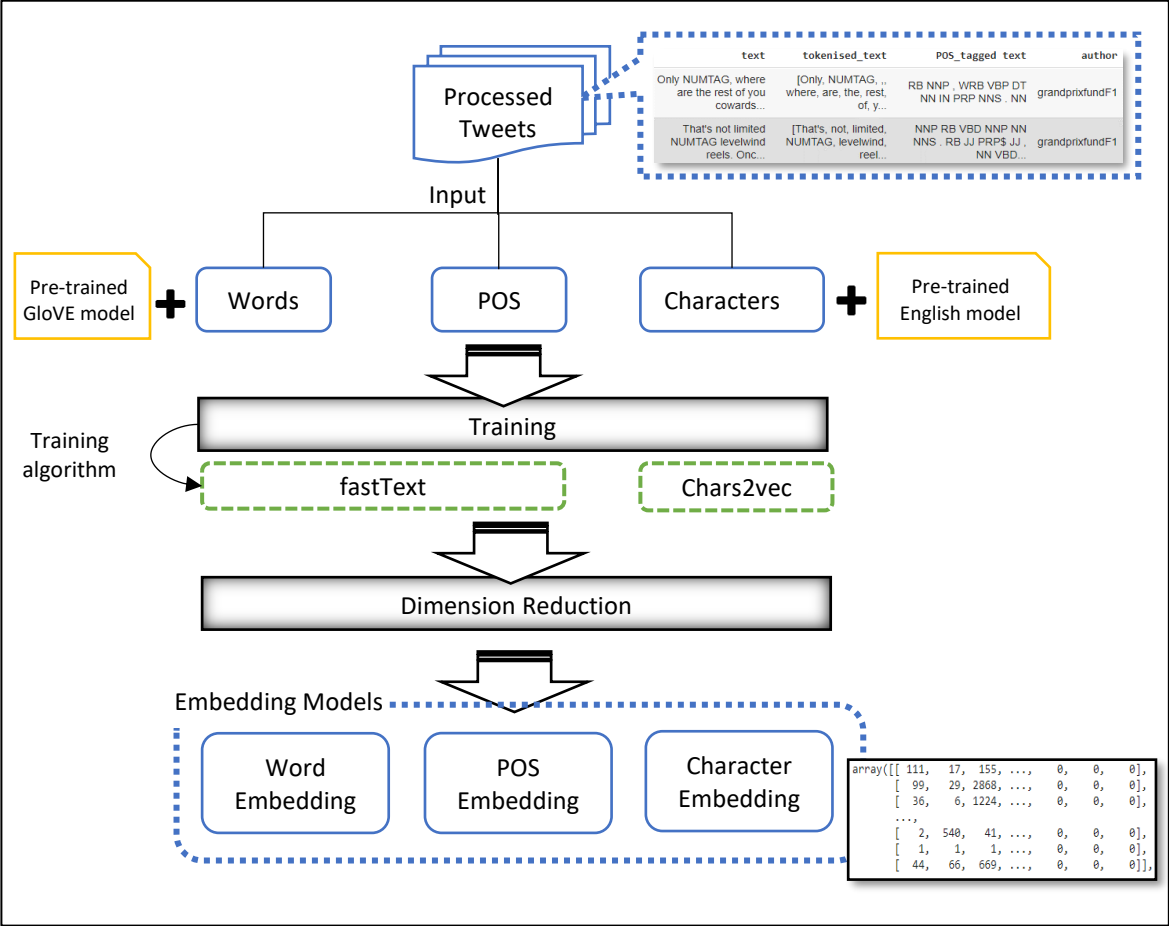


Figure 3.14: Embedding-based representations of tweets

The embeddings are the modern way of representing words in terms of fixed-length, continuous, and dense feature vectors. Embedding representations capture hidden information about a language, like word analogies or semantic. It also helps to improve the performance of text classification. There are three levels of embeddings that are explored, i.e., word, POS and character embeddings. Both word and POS embeddings employed the

same algorithm for training, fastText (Bojanowski, Grave, Joulin, & Mikolov, 2017), while character embedding model is trained using Chars2Vec model (Intuition Engineering, 2019).

- i. Word Embedding

Word embedding imposes a semantic model by collating words related by sense with the affine embedding vectors. These semantic models represent contextual proximity of various words: such a model is trained on large corpora of existing text such as encyclopaedias, new texts, or literature. The result is that proximal vectors represent the words appearing in similar contexts in these texts. In this thesis, fastText pre-trained English word vectors (Bojanowski et al., 2017) are used.

- ii. Character Embedding

Word embedding contains some information about the sense of words extracted while training on large text corpora like fastText, with a fixed vocabulary usually lacking some rarely used words. If there are important words in a text that do not belong to the model's vocabulary, then the model is incompetent to interpret the important words semantically. The problem often arises in text written in OSN like tweets where there is high usage of internet parlance (i.e., LOL), repeated punctuation (i.e. !!!), and typos that produce “new” words (i.e., *Plis*) without any collated embedding.

Chars2vec model is used for character embedding representation to solve the word embedding model's incompetency. It creates word embedding based on their spelling and collates similar vectors to similarly spelt words. The model represents each sequence of symbols with a fixed-length vector. The distance metric between vector represents the similarity in words spelling.

Figure 3.15 depicts an example of a projection of Char2vec embedding vectors on a plane to show similar vectors to similarly spelt words. A pre-trained Intuition Engineering English model with 300 dimensions is used for training the tweet. The `c2v_model.vectorize_words` function will return an array of shape (n_tweets, 300) with word embeddings. 300-dimensions of the pre-trained model is chosen because this model has obtained the highest accuracy compared to other dimensions.

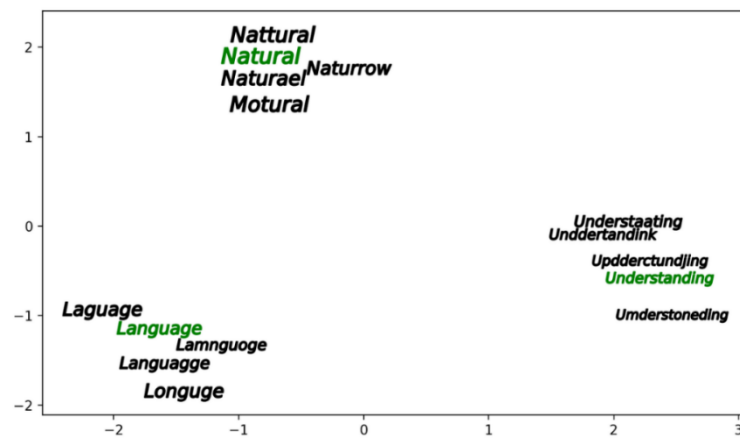


Figure 3.15: A projection of embedding vectors of similar words using Chars2vec model (Intuition Engineering, 2019)

iii. POS Embedding

For POS embedding, there is no pre-trained corpus as word embedding. Thus, the tweet corpus was tagged using NLTK pos_tag and self-trained using the skip-gram learning model. Skip-gram architecture predicts the context of a given word. In this case, the skip-gram model (Mikolov et al., 2013) learns vector representation for each POS tag (w) belonging to Penn Treebank POS tagset $\{1, \dots, W\}$.

The representations are trained to predict POS tags appearing within the context window of a given centre POS tag. Given the tweets that have been POS tagged and can be thought of as a sequence of POS tags $w_1, w_2, \dots w_T$. The skip-gram model maximises the log-likelihood, i.e., the probability of a context POS tag given a centre POS tag.

$$\sum_{t=1}^T \sum_{c \in C_t} \log p(w_c | w_t) \quad \text{Equation 3.3}$$

where, C_t is the context window around the centre POS tag w_t . The probability of the context word given the centre word is softmax over the scoring function as see below:

$$p(w_c | w_t) = \frac{e^{s(w_t, w_c)}}{\sum_{j=1}^W e^{s(w_t, j)}} \quad \text{Equation 3.4}$$

In Equation 3.4, s is referring to the scoring function, which is the dot product of the POS tag vectors of the centre and the context word. In generating the POS level embedding model, the genism Word2Vec library is accessed. The parameters are set as follow:

- Dimensionality size of the vector: 100
- Window: 5
- Iteration: 100

iv. Mix Embedding

Mix embedding comprises three levels of embedding representations (word, POS, and character), and the embedding vectors are concatenated. Such as word vector with 300-d, POS with 100-d, and character with 300-d, and when they joined, it will produce an array of shape (n_tweets, 700) mix embeddings.

3.3.2 Classification of Tweets' Authors

After constructing text representation models, classification algorithms are used to classify the tweets based on their respective authors. The algorithm produced an AI model that represents the knowledge it has learned from the vector model. The learned AI model later is used to predict the most plausible author of an anonymous tweet. There are two approaches in classifying the tweets; machine learning (ML) employing traditional ML algorithms and deep learning (DL) employing neural network algorithm.

The tweets will undergo feature extraction process if the ML algorithms do the classification. In DL, tweets in embedding forms do not undergo feature extraction when employing a neural network algorithm during classification. Four classifiers will be deployed: Naïve Bayes, Random Forest, Support Vector Machines (SVM), and Convolutional Neural Network (CNN) to build AI models.

The selection of classifiers related to the chosen representation models. TF-IDF representations are trained using Naïve Bayes, Random Forest and SVM due to the robustness of these algorithms against noisy and sparse feature representation. Meanwhile, embedding representations are trained on CNN because CNN generally works well with data with spatial relationship and dense vector.

a. Naïve Bayes

Naïve Bayes builds a probabilistic model for each authorship class based on the training data of that class, y . The probabilities of features, $P(x_i)$, are then computed by multiplying all the features to give the probability of test data. The highest probability, $\hat{y}$, among all authors (classes) is the most plausible author of the test data or anonymous tweet. The following equation shows the classification rule of Naïve Bayes (Kowsari, et al., 2019):

$$\hat{y} = \arg \max_y P(y) \prod_{i=1}^n P(x_i|y) \quad \text{Equation 3.5}$$

The different Naïve Bayes classifiers may differ by the assumption they make regarding the distribution of $P(x_i|y)$. Multinomial Naïve Bayes is implemented (Savoy, 2020). The classifier lets us know that each $P(x_i|y)$ is a multinomial distribution.

The distribution is parametrised by vector $\theta_y = (\theta_{y1}, \dots, \theta_{yn})$ for each class y , where n is the number of features (vocabulary count) and θ_{yi} is the probability $P(x_i|y)$ of feature i appearing in a sample belonging to class y . The parameter θ_y is estimated by a smoothed version of maximum likelihood, i.e., relative frequency counting:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n} \quad \text{Equation 3.6}$$

Where $N_{yi} = \sum_{x \in T} x_i$ is the number of times feature i appears in a sample of class y in training set T , and $N_y = \sum_{i=1}^n N_{yi}$ is the total count of all features for class y . However, at times the test data contain features that have not been seen in the training data.

A problem may arise to Naïve Bayes classifier when the stated situation happened as some probabilities yield zero-result since none of the training data falls in the range. The zero counts will have a zero probability and leave the Naïve Bayes classifier impotent to predict a class for the test dataset.

b. Random Forest

A random forest is an ensemble of many classification trees. The random forest model comprises a set of decision trees; each tree is trained using random subsets of features. Each tree consists of three types of structures, which are: - leaves, interior nodes, and

branches. The leaves of the tree represent the classes (authors), the interior nodes represent the features (vocabulary), and the branches represent the values held by the feature (which in this case is the TF-IDF value).

To classify a tweet, the author of that tweet is tracked from the tree root until it reaches the leaf where the class is located. The construction of a tree is selected based on the nodes and branches that produced the best split. Following are the Random Forest parameters set for the experiment during implementation:

- Gini impurity is used as a default function to measure the quality of the split
- The number of trees produced in the forest is set to 100
- The maximum depth of the tree is set to None. The nodes will expand until all leaves are pure or contain less than the minimum number of samples required to split.
- The minimum number of samples required to split an internal node is set to 2
- The minimum number of samples required to be at the leaf node is set to 1
- Bootstrap samples are used in building the trees

Although Random Forest is less impacted by noise and reduces overfitting, the classifier requires more computational power and resources as it produces many trees. Additionally, the Random Forest classifier also demands a longer time in training data as it generates many trees.

c. Support Vector Machines

SVC (C-Support Vector Classification) is used for classifying the tweets where the implementation is based on LibSVM (Savoy, 2020). LibSVM solves the multi-class problem; that is why it is used to handle multi-classes author identification problems. The multi-class support is handled according to a one-vs-one strategy. This strategy constructs a classifier for each pair of classes.

LibSVM is a faster implementation as it applies techniques which are: - caching and shrinking. Caching means that the earlier computed values are stored in memory so that re-computation is unneeded. Shrinking technique temporarily eliminates variables that have reached a predefined lower or upper bound. Therefore, these values are not used in subsequent computations.

When training an SVM classifier, several parameters can be set. One of the parameters that are important in SVM is the type of kernel used. SVM kernel is a function that takes low dimensional input space and transforms it into higher dimensional space. It helps in converting the inseparable problem into a separable problem. For the experiment, a linear kernel is set as this kernel is preferable when the number of features is extensive. Other parameters are set as followed:

- C as a regularisation parameter is set to 1
- Gamma as a kernel coefficient is set to 0.001
- Shrinking is set to True
- Cache size is set to 200
- Decision function of shape using *OVR* (one-vs-rest) with the shape of $m \times n$ (n_samples, n_classes)

Although the SVM classifier is very advantaging on high dimensional spaces like n-grams for short messages, SVM may not perform well when the data has more noise. Moreover, SVM's performance may decrease if the number of features for each data exceeds the number of training samples. Like in our case, the number of features is extensive and sparse; thus, this will make SVM underperform.

d. Convolutional Neural Network

For the AI task, the input is the tweet, which is represented as a matrix. Every matrix is a vector of text (word/character/POS tags) indexed within the limit of the top words (or characters or POS tags), which has been defined. Each matrix is padded and represented either word, character, or POS tag matrix, and the concatenation of those three matrices. The matrix is padded so that the kernel filter and stride can fit the matrix well. The matrix is padded to a maximum length in words (or characters or POS tags).

CNN is based on a single channel, where the matrix representation is a concatenation of its word (or character or POS tag) embedding generated by either using a pre-trained model (for word), char2vec model (for character) and self-trained (for POS tag). The CNN model's construction is based on Shrestha et al. (2017), in which the model is modified in several ways.

The 1D convolutional layer slides 100 filters of various window sizes ($n=3,4,5$) over the input channel (embeddings). The filters generate a new feature for a window of words (or characters or POS tags). The application of filter over each possible window of words (or characters or POS tags) in the sentence produces a feature map. Max-over-time pooling condenses the feature vector to its most important feature by taking its maximum value. Figure 3.16 is an illustration of the CNN hyper-parameter for text classification.

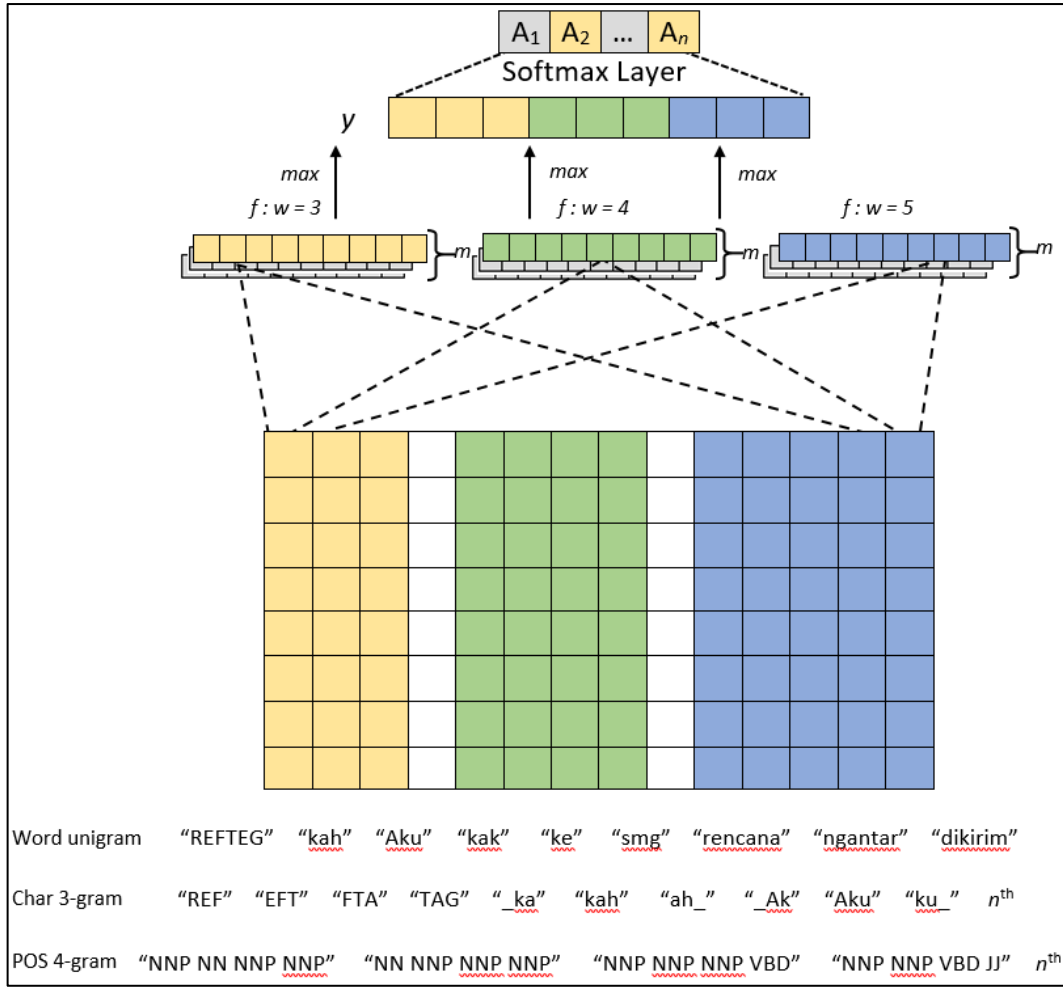


Figure 3.16: Embedding N-grams CNN architecture

As shown in Figure 3.16, the input of padded sequence n-grams is passed through the Embedding Layer, Convolutional Layer, and lastly, to the fully connected *softmax* layer. The maximum length, l , of the n-gram sequences determines the input size. The Embedding layer receives the vocabulary size or the number of n-grams, the size of dimension, and the maximum length of the tweets. Then, the embedding layer produce a matrix, $C \in R^{d \times l}$, where the rows represent the sequence of n-grams, while the columns represent their embedding c_j of position j .

Next, in the convolutional layer, a convolutional filter, $H \in R^{d \times w}$, is applied to a portion of C , where w is the width of the filter. The resulting matrix, O , is used as input to a *relu* function g , along with a bias term b to produce feature representation f for the tweets.

$$O = H \cdot C[i : i + w - 1] \quad \text{Equation 3.7}$$

$$f = g(H \cdot C[i : i + w - 1] + b), f \in R^{l - w + 1} \quad \text{Equation 3.8}$$

In the *1D* convolutional layer, different widths w are used to capture morpheme-to-word patterns. The *maximum pooling* layer condensed the feature maps f , to obtain y_k , the maximum value of each feature map f_k , where m is the number of feature maps.

$$y_k = \max_i f_k[i], k = 1 \dots m \quad \text{Equation 3.9}$$

Pooling is necessary to represents the text's most crucial features, regardless of their position. After pooling and merging the feature representations y_k , a compact representation of the text is yielded. To convert the three-dimensional output to two-dimensional for merging, a *flatten* layer is employed.

Lastly, the dense feature matrix is passed to the fully connected layer of *softmax* layer. Table 3.3 describes the combination of hyper-parameters for the three layers of CNN. Additionally, *batch normalization* layer is added right after the convolutional layers to maintain the mean output by normalizing the output values using the mean and standard deviation of the batch of inputs.

Table 3.3: CNN architecture hyper-parameters

Layer	# of layers	Hype-parameters	
Embedding	1	l d	max_length 300
Convolutional 1D	3	m w <i>Pooling</i>	[128,128,128] [3,4,5] Max1D
Fully connected	1	# of authors	1290

In Table 3.3, the hyper-parameters, including *Adam* optimizer is used to compile the CNN model, and *sparse_categorical_crossentropy* is used to calculate the loss. Apart from that, *SparseCategoricalAccuracy* is used to calculate the accuracy for 100 epochs.

In this thesis, CNN is chosen rather than sequential-based models (i.e. LSTM and GRU) because it is computationally efficient for OSN text as it can parallelize computation across multiple filters. This makes it faster to train on large datasets, which is beneficial for working with a vast amount of OSN text. Also, OSN text is often noisy and contains a lot of irrelevant information which sequential models may struggle to make accurate predictions as the models better at capturing long-range dependencies. While, CNN is better at capturing local patterns in text, which makes it more suitable for dealing with noisy data.

3.4 Evaluating the Learned Models of Author Identification

The performance of the learned model will be measured to determine how good the model is. *K*-cross validation is used to validate the classifiers.

3.4.1 Evaluation Metrics

Four evaluation metrics are used to measure the models' performance: accuracy, Precision, recall, and f-score. There are various metrics to evaluate the classification performance. The following are the evaluation metrics used in the experiments:

- Mean accuracy
- Precision (P)
- Recall (R)
- F-score (F1)

The explanation of each metric is explained in detail below. The metrics calculations are necessary to assign the result of each classification of the following four types of results:

- A true positive (TP) – the classifier has identified a positive class data point as positive
- A false negative (TN) – the classifier has identified a negative class data point as negative
- A true negative (FN) – the classifier has identified a positive data point as negative
- A false positive (FP) – the classifier has identified a negative class data point as positive

A confusion matrix is constructed from the frequencies of the result, as shown in Table 3.4.

Table 3.4: Confusion Matrix

		Predicted Class	
		Yes	No
Actual Class	Yes	true positive (TP)	false negative (FN)
	No	false positive (FP)	true negative (TN)

According to (Savoy, 2020) on page 64, **Precision** is defined as follow:

$$Precision(P) = \frac{TP}{TP+FP} \quad \text{Equation 3.10}$$

Precision is intuitively the ability of the classifier not to label as positive a sample that is negative.

Recall is defined as follow:

$$Recall(R) = \frac{TP}{TP+FN} \quad \text{Equation 3.11}$$

Recall is intuitively the ability of the classifier to find all positive samples.

F-score is defined as follow:

$$F - score(F_1) = \frac{2 \times R \times P}{R+p} \quad \text{Equation 3.12}$$

F_1 can be interpreted as a weighted average of Precision and recall. F_1 reaches its best value at 1 and worst at 0. The relative contribution of Precision and recall to F_1 are equal.

Accuracy is defined as follow:

$$Accuracy = \frac{TP}{TP+TN+FP+FN} \quad \text{Equation 3.13}$$

Accuracy is the proportion of true results among the total number of data points examined.

3.4.2 Validation Technique

For training and testing, k cross-validation technique is implemented. K cross-validation is chosen because the classification problem has limited data to work with; it can be challenging to provide enough data for disjoint training and testing sets. In the forensic setting, it happens to have a small amount of data to produce an authorship model. K cross-validation provides more meaningful results by using all of the data set as both training and testing data. Figure 3.17 illustrates how cross-validation works.

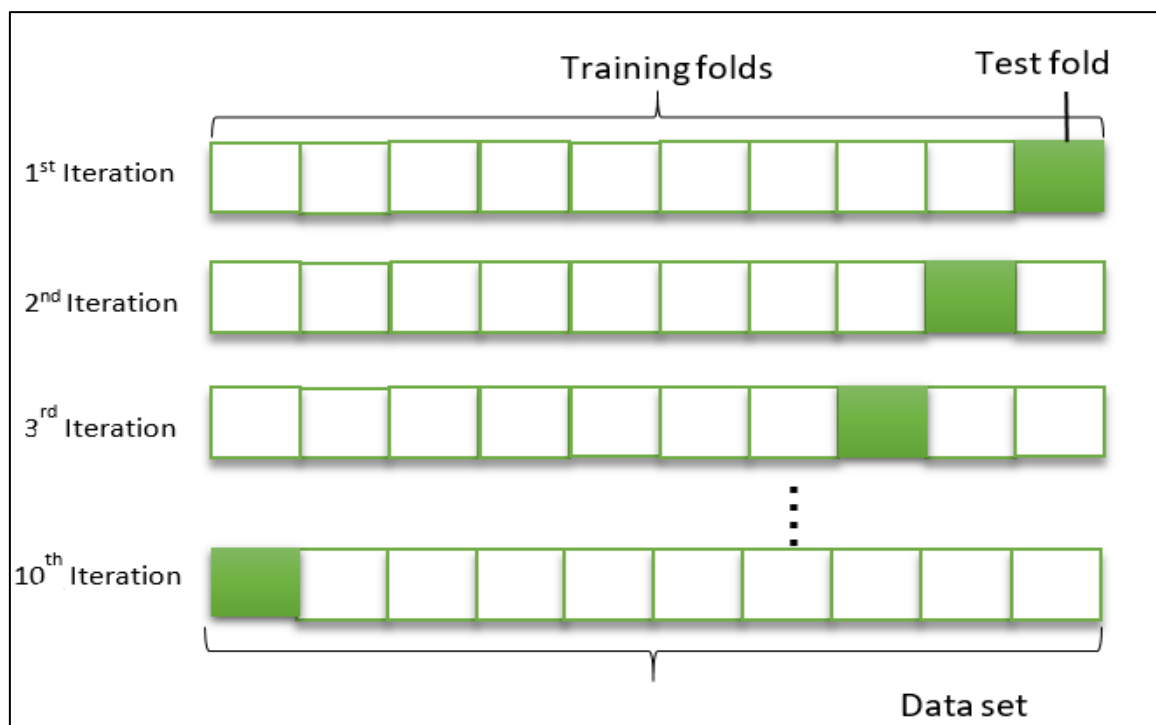


Figure 3.17: k cross-validation technique

Based on Figure 3.17 in k cross-validation ($k=10$), the data is split into 10 folds, equal in size. A classifier learned from 10-1 folds of data, with the remaining fold being used as a test set. The process is then repeated so that each fold is held out for testing. The results of the classification from the 10 tests are combined to calculate the overall results for the data set.

3.5 Summary

This chapter outlines the implementation of the proposed AI in OSN text to facilitate SMF. It covers the entire workflow, including data preparation, data extraction from Twitter, pre-processing, and text analysis. The classification phase explores two representation models (TF-IDF and embedding) and four classifiers (Naïve Bayes, Random Forest, SVM, and CNN) to build the best AI model for analysing social media text. The evaluation involves validating the learned model using k cross-validation and assessing performance using accuracy, precision, recall, and f-score. The chapter also includes a Chapter 3 mind map in Appendix C. Next Chapter 4 focuses on the experimental setup for authorship identification of short text.

CHAPTER 4

RESULT AND DISCUSSION

AI task has gained increasing popularity since it is associated with critical forensic applications. Thus, there is a significant need in AI, especially for SMF that can exploit the context of short online messages. To adapt AI for SMF, there is a demand for finding the appropriate machine learning components that subsist with the diversity of social media parlance that keeps evolving deployed therein. Hence, the evaluation of the proposed AI workflow for SMF, as in Chapter 3, is presented in this chapter. This chapter aims to determine whether the proposed AI system's workflow is suitable for identifying the author of social media text for forensics study. The task is accomplished by evaluating the workflow on a series of experiments. Close-set attribution setting often fits the forensics applications where specific individuals can access certain resources and have knowledge on specific issues. Hence, the evaluation of AI performance is conducted based on the closed-set setting, which corresponds to the forensics applications.

4.1 Tweets as Test Corpus

Twitter are accessible using the Twitter API that allows data to be collected easily. Other platforms are hard to obtain because of privacy issues, and access to the data is prohibited. The test corpora used, and the evaluation configurations will be explained in the sub-sections below.

4.1.1 English Tweets

English tweets are collected using a feedlist (refer Appendix A) of profane English words retrieved from a website. The interest is towards tweets that contain vulgar or profane

words to mimic the offensive contents post by potential offensive users on Twitter. 50 authors are considered to evaluate the proposed workflow because, in forensic linguistic scenarios, usually limited number of suspect authors are implied as mentioned in Section 3.2.1b.

4.1.2 Under-resourced Language Tweets

In addition to English, the proposed work also tests the use of under-resourced language (U-RL) tweets. Malaysia, particularly Borneo, has numerous indigenous languages with stable community speakers. *KadazanDusun* is one such language, widely spoken in Sabah, Malaysia. Although *KadazanDusun* is actively used in online communication, it lacks a writing system. Therefore, Natural Language Processing (NLP) is crucial for analysing U-RL text, especially for investigating cyberbullying. *KadazanDusun* tweets are used in this study to evaluate the accuracy of AI for sentiment analysis. Acquisition of *KadazanDusun* tweets is carried out using a feedlist of *KadazanDusun* words. Unlike English, profane words are rare in *KadazanDusun*, and speakers often express cursing through euphemisms or taboo words. The process of acquiring tweets for *KadazanDusun* is similar to that of English, involving 50 prolific users.

a. Pre-processing of *KadazanDusun* Tweets

After crawling tweets, the pre-processing stage is done, but it is slightly different from English tweets. There are mixed of English words being found that in most tweets of *KadazanDusun*. Any English words that occur in the *KadazanDusun* tweets are eliminated. The reason is, to maintain the originality and quality of the *KadazanDusun* language in the tweets.

The elimination of English words in *KadazanDusun* tweets is done by searching and matching the words with English dictionary from NTLK. If the words matched with the dictionary's vocabulary, then the words will be removed from the tweets. Other than that, two other English corpora with less formal languages are implemented to match the English words in the tweets which are:

- i. Naval Postgraduate School (NPS) chat corpus: A corpus of instant messaging chat session which contains over 10,000 posts from various age-specific chatrooms.
- ii. Webtext corpus: A collection of web text includes content from a Firefox discussion forum, conversations overheard in New York, the movie script of *Pirate of the Caribbean*, personal advertisement, and wine reviews.

After completing data cleaning, continue with tokenisation and POS tagging using default POS tagger same as English tweets. Finally, the tweets are in the form of processed text and ready for the next phase.

4.2 Evaluation Strategies

The evaluation is carried out to answer the research questions (RQs) and accomplish the research objectives (ROs), as stated in Chapter 1. Figure 4.1 depicts the flow diagram from RQs to experimental implementation.

* *RQ = research question, RO = research objective, E = experiment*

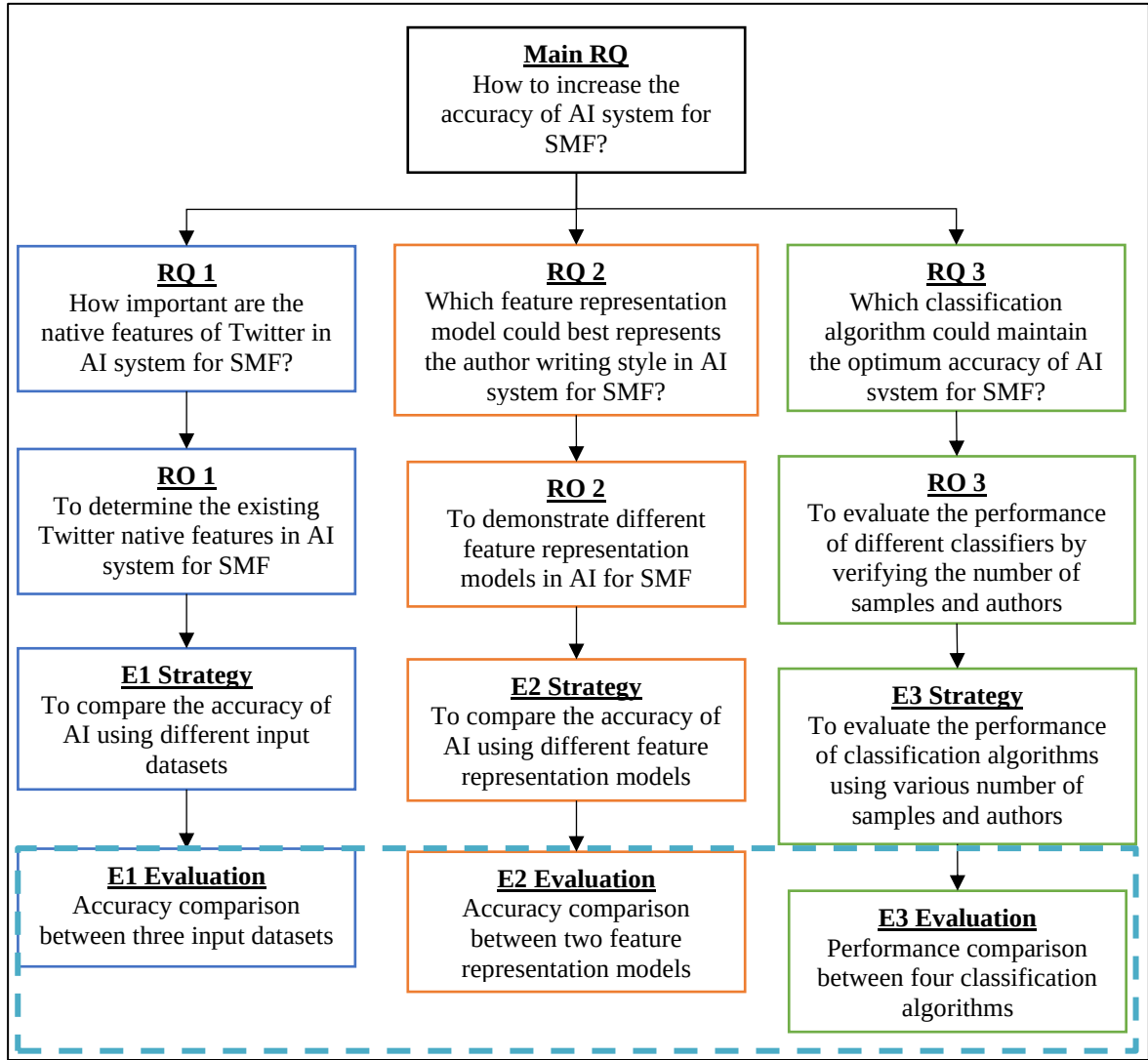


Figure 4.1: The sequence of answering the research questions

Several configurations are set up in the series of evaluations to answer the RQs., A series of individual tests are performed on each component to access each AI component as crafted and configured in Table 4.1. There are three sets of experiments designed to answer the research questions of this study.

Table 4.1: The experimental strategies and configurations

		Configurations			
		Strategy	Samples/ Classes	Features	Algorithm
Experiment 1 (Baseline)	RQ 1: How important are the native features of Twitter in the AI system for SMF?	Prepare different datasets:	500 tweets per author/ 10 authors	N-grams: Word unigrams + Char 3-grams + POS 3-grams	SVM
	Experiment Strategy: To compare the accuracy of AI using different input datasets	a) <i>S+N (stylo+native)</i> b) <i>S (stylo)</i> c) <i>N (native)</i>		Embedding- based: Word + Char + POS	CNN
Experiment 2	RQ 2: Which feature representation model could represent the author's writing style in AI system for SMF?	Implement different feature representations:	(S+N dataset) 500 tweets per author/ 10 authors	N-grams: • Word level = ($n \in 1,2,3,4,5$) • Char level = ($n \in 3,4$) • POS level = ($n \in 3,4$)	SVM
	Experimental Strategy: To compare the accuracy of AI using different feature representation models	a) N-grams representation b) Embedding-based representation		Embedding-based: • Word level • Char level • POS level	CNN
Experiment 3	RQ 3: Which classification algorithm could maintain the optimum accuracy of AI system for SMF?	Implement different classification algorithms:	(S+N sub-dataset) n authors = (10, 20, 30, 40, 50) n samples = (200, 250, 300, 500)	N-grams: Word unigrams + Char 3-grams + POS 3-grams	NB, RF, SVM
	Experimental Strategy: To evaluate the performance of classification algorithms using various number of samples and authors	a) Naïve Bayes b) Random Forest c) SVM d) CNN		Embedding- based: Word + Char + POS	CNN

Based on Table 4.1, there are three different data sets are prepared for Experiment 1 by manipulating the normalisation process in pre-processing. The three data sets are:

- i. *S+N* (stylometry + native features) dataset - native features of Twitter are included together in tweets in the form of standard tags (presence of native features along with text data)
- ii. *S* (stylometry) dataset - all native features are removed in the tweets (absence of native features in the text data)
- iii. *N* (native feature) dataset - all text in the tweets is removed leaving only the native features (presence of native features without text data)

The hyper-parameter configurations for each classification algorithms are described below in Table 4.2 below.

Table 4.2: Hyper-parameter configuration for classification algorithms

Classification Algorithm	Hyper-parameter configuration
SVM	Class: SVC Value of C: 1 Gamma: 0.0001
CNN	Num. of filters: 100 Window size: 3,4,5 Dropout value: 0.3
RF	Num. of tree: 100 Criterion: GINI impurity Random state: 0
NB	Class: Multinomial NB Alpha: 1.0 Class prior probability: True

For all experiments (ref Table 4.2), the data set is partitioned into training and test sets via 10-fold cross-validation. Average classification accuracy is reported as a summary statistic. After designing the strategies, the experiments are conducted based on the classification algorithm configurations to acquire the results performance. Later, the results are compared, and in-depth analyses are discussed.

4.2.1 Evaluation on the Importance of Twitter Native Features

Experiment 1 is designed to access the input data by analysing the importance of Twitter native features in stylometry to identify the author of social media text. A comparative study is done to compare the accuracy of AI, based on the involvement of native features in the tweets.

4.2.1.1 Setup for Experiment 1

The experiment required preparations of three different datasets to be tested. The production of three different datasets is done by manipulating the normalisation process in Phase 1, as shown in Figure 4.2.

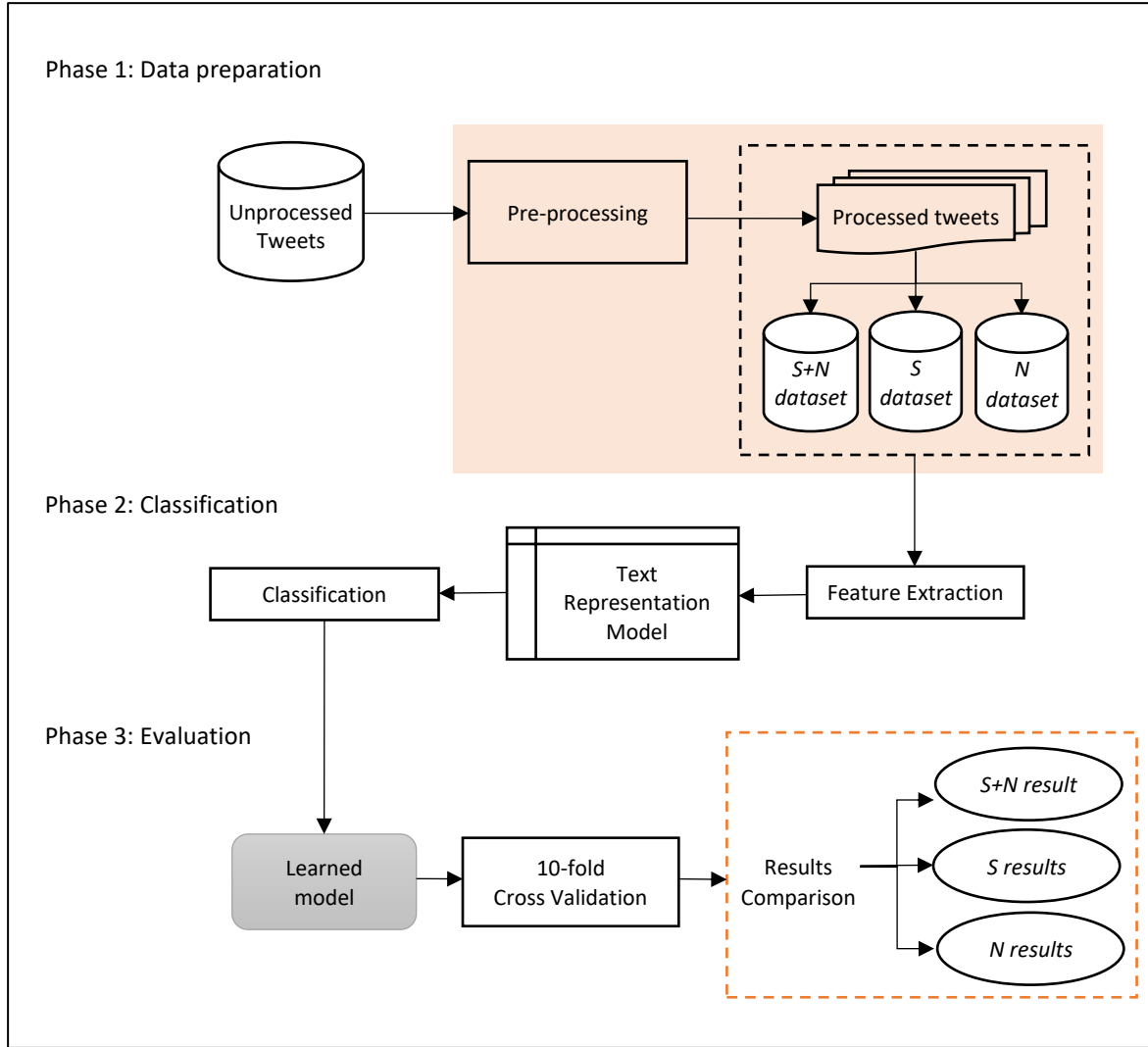


Figure 4.2: Experimental design for Experiment 1

The importance of Twitter native features is examined on the accuracy of AI system, by conducting experiment that compares three data sets. Table 4.3 shows an example of an English and *KadazanDusun* tweet after being handled differently in pre-processing task, specifically during normalisation to output as three different data for each dataset. The example shows how the datasets differed from each other.

Table 4.3: Example of a pre-processed tweet between three different datasets

		English	KadazanDusun
Original tweet		You're a complete dipshit! Self made I'm sure! https://t.co/Si7qZ802lC	@MAHA_namanya Klau mnuman, ada satu baru . Tp sa nda brp suka. Ahahaahah itu ja laa yg okay .
Pre-processed tweet	S+N tweet	You're a complete dipshit! Self made I'm sure! URLTAG	REFTAG Klau mnuman ada satu nda brp suka Ahahaahah itu laa yg
	S tweet	You're a complete dipshit! Self made I'm sure!	Klau mnuman ada satu nda brp suka Ahahaahah itu laa yg
	N tweet	URLTAG	REFTAG

Referring to Table 4.3, the features are represented with TF-IDF value of word unigrams, character (char) 3-grams and Part-of-speech (POS) 3-grams combinations for SVM. The selection of features and classifier is inspired by Banga and Mehndiratta (2017), wherein their study the combinations of word, character and POS N-grams with SVM proved to produce optimum accuracy in attributing tweets.

As an alternative approach to TF-IDF N-grams model, embedding-based representations using CNN as the classification algorithm is implemented. The implementation of word embedding is inspired by the studies done by Posadas-Durán et al. (2017); Hitschler et al. (2018) using word and POS data. As for character embedding, a study done by Shrestha et al. (2017) paved a way to explore the strength of characters using CNN in attributing author for short text by yielding high accuracy.

4.2.1.2 Result Analysis of Different Datasets

Experiment 1 discusses the result analysis of Twitter native features impact towards AI for SMF, whereby the experiment is conducted by comparing the results of three distinct datasets, $S+N$ dataset, S dataset and N dataset to examine the impact of Twitter native features on the accuracy of AI system.

Figure 4.3 reports the accuracy results of AI for three mentioned datasets using SVM and CNN. The results composed of English and *KadazanDusun* tweets with 500 samples provided for every ten authors.

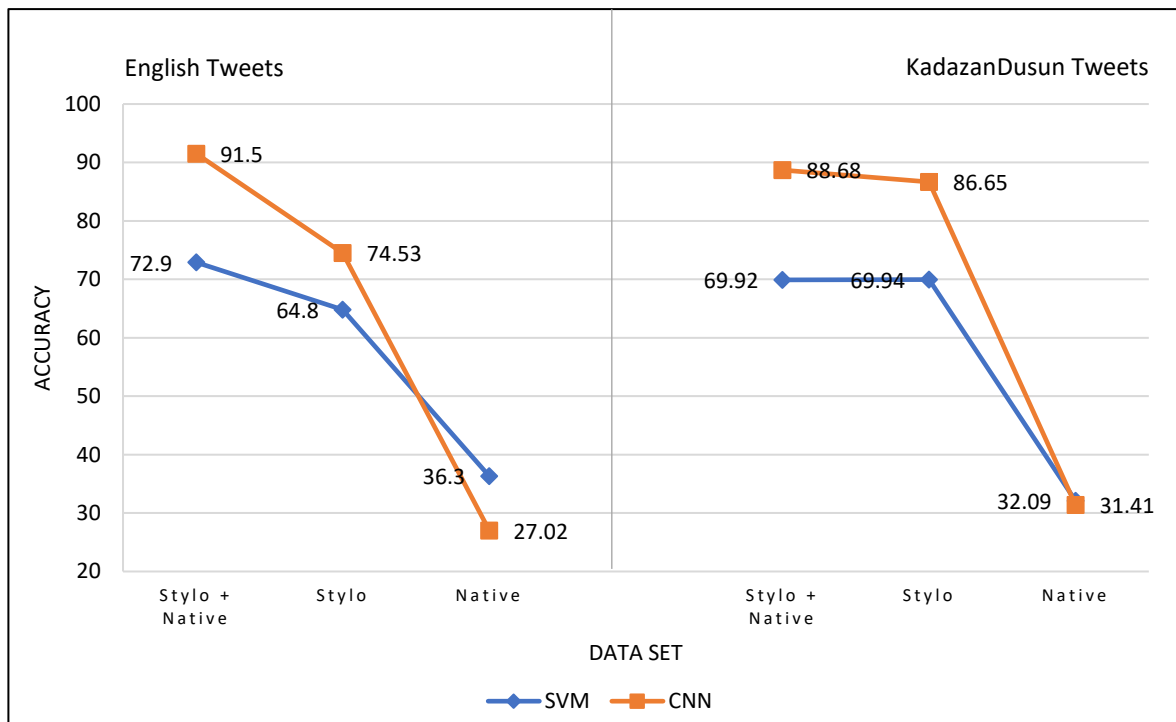


Figure 4.3: Accuracy comparison of three different datasets for English and *KadazanDusun* tweets

Overall, both corpora's accuracy shows decline trends across the datasets from $S+N$ to N data set (ref. Figure 4.3). The accuracy rates for $S+N$ were significantly higher than S and N datasets. Both corpora show the highest accuracy reported in $S+N$ with CNN preceded SVM with over 18%. Although there is a slight difference in the accuracy of classifiers for all three data sets, AI system performs better with the presence of Twitter native features in stylometry.

Compared to the S data set, the impact of native features presence in $S+N$ data set seems noticeable in English corpus where the performance boosts up to 16.7% for CNN and 8.1% for SVM. The increment is explainable as besides stylometry, native features play a crucial role in adding information in the writing style of authors in social media, Twitter. Provided with only 280 characters limitation in each tweet, AI performance for social media short text managed to produced accuracy up to 72% when native features are included using SVM and 90% using CNN.

Twitter native features, i.e., the URLs, user references and hashtags, are taken into account by normalising the features to standard tags to add information into the stylometry through $S+N$ data set. However, if the features are disabled or excluded from the tweets as in S data set, the accuracy dropped gradually. The finding proved that the presence of native features facilitates the stylometry in distinguishing the writing style of the authors for English tweets.

Notice that, *KadazanDusun* corpus recorded slight fluctuations from $S+N$ to S dataset with a slight increase of 0.02% (SVM) and a gentle drop of 2.03% (CNN). The presence of native features had failed to create a powerful impact on AI accuracy in *KadazanDusun* corpus compared to English. The situation can be explained by the state of the data

(*KadazanDusun* tweets) itself. Two hypotheses could account for the fluctuated trend circumstance. First, excessive removal of stylometry information in *KadazanDusun* tweets is the immense contribution to this circumstance. It is essential to state that the *KadazanDusun* tweets are scarce to be crawled.

Moreover, most *KadazanDusun* tweets contain a mixture of foreign language. Foreign words are removed from the tweets to uphold the indigeneity of *KadazanDusun*. The removal resulted in stylometry information abatement where the number of words and characters in each tweet shrinks. Second, POS tagging for *KadazanDusun* tweets is done using default English POS tagger where the tokens are tagged based on the English Penn Treebank tagset. Although it is inappropriate, yet the reason is that every NLP tools and processing steps made in English need to be done correspondingly to *KadazanDusun*. The decision is made to make sure that the result for both languages is comparable. Thus, the tagging process indirectly affected the result of *KadazanDusun* tweets.

There is perceived failure of the *N* dataset to provide stable accuracy. In Figure 4.3, the accuracy for both languages is dropped steeply below 40% of the accuracy rate. This dataset showed the lowest accuracy of 27.02% (CNN) in English with a little raise of 9.28% (SVM). For *KadazanDusun* corpus, both classifiers remain steady at between 32% and 31%. *N* dataset recorded the worst because it preserved only native features and disregarded the authors' stylometry information (text and punctuation). Despite relying on native features solely, stylometry contains typically crucial information on the authors' writing style. Both classifiers can classify well if sufficient information is provided or else the writing style remains indistinguishable.

Most previous experiments choose to exclude the native features from the tweets. In their experiment, Banga and Mehndiratta (2017) cleaned the tweets gathered by removing the native features of Twitter, including URLs, user references, and punctuations. Using combination features of word, character and POS N-grams, the results they gained for SVM is 69.83% and MLP (neural network) with 64.5%. Compared to this study, overall, the accuracy gained is less than 21.67% (SVM) and 7.9% (CNN). In another study conducted by Igawa et al. (2015), they erroneously state that Twitter URLs and retweets were removed since these data do not infer any information about the user's writing pattern.

However, the findings from Experiment 1 proved that Twitter native feature does give an impact on the accuracy of AI especially for short social media text. Experiment 1 concludes that the strategy of comprehending Twitter native features as a part of stylometry offers a competitive performance with significantly improving the accuracy of identification for social media authors. Especially when dealing with short and limited source of information in social media text like Twitter (short yet full of Internet parlance and punctuation involvement), the inclusion of the application's native features plays a crucial role in discriminating the authors.

4.2.2 Evaluation of Different Feature Representation Models

Experiment 2 attempts to access the second component of AI, which is the features. The strategy of Experiment 2 composed of two parts. Besides comparing the feature representation models, individual test for each feature types (word, character and POS level) is examined to access the usefulness of the features.

4.2.2.1 Setup for Experiment 2

As a whole, Experiment 2 focuses on the feature types and its representation model. Between the two representation models, which one that befits AI for SMF. The illustration of Experiment 2 strategy is highlighted in Figure 4.4.

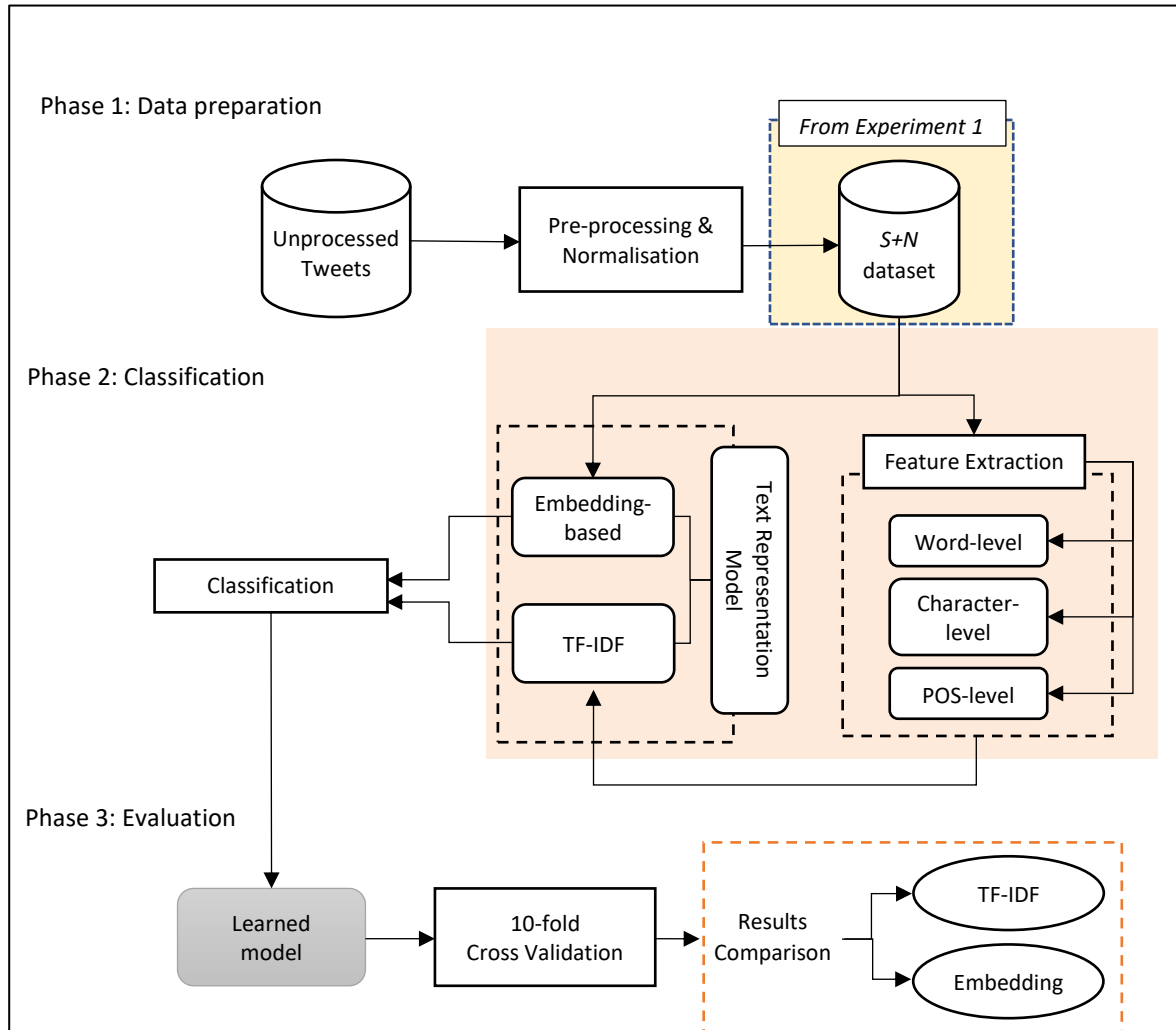


Figure 4.4: The design of Experiment 2 strategy

According to Figure 4.4, the idea of Experiment 2 is to represent the training documents as a numerical vector and use the classification algorithm to classify the training vector to its known author. Two representation models are implemented and examined to find the best-fitted model to represent short social media text (tweets). The first model is Term Frequency-Inverse Document Frequency (TF-IDF) representing N-grams features based on TF-IDF value.

Different N-grams feature sets are generated: word-level N-grams (for $n \in \{1,2,3,4,5\}$), character-level N-grams (for $n \in \{3,4\}$), and POS N-grams (for $n \in \{1,2,3\}$). To access the usefulness of the feature types, independent tests are performed to each level of N-grams. Rocha et al. (2017) has introduced the strategy as a scalability assessment for AI in social media; thus, the strategy is implemented in this study.

However, N-grams problems is that, as more samples and classes present in the classification, vocabulary size will explode and produce sparse data (large feature space). An alternative way of representing the document is explored to overcome the problem. Embedding-based model is tested and compared to TF-IDF as this model resulted in dense representation. For embedding-based model, word, character, and POS data are explored using word embedding (for word and POS input) and character embedding (for character input).

So, after obtaining the representations of tweets, both models are passed to the classifier. In this experiment, two base classifiers are used to test the models which are SVM and CNN. The classifiers are selected based on two reasons:

- i. SVM is robust in handling sparse data but can SVM performs in classifying tweets using embedding based model?

- ii. CNN is proven powerful in producing high accuracy on the embedding-based model but can CNN still maintain its performance using sparse vector representation?

The strategy of Experiment 2 is inspired based on the research done by (Vajjala & Banerjee, 2017), in their study, they utilised long-text data containing essays and speech transcript tested on the TF-IDF and embedding model.

4.2.2.2 Result Analysis of Different Feature Representation Models

Experiment 2 can be splitted into two observations. First, observing the performance of different feature types for each model. Second, observe the performance of models based on different classifiers. Validations are done using 10-fold cross-validation. The results of Experiment 2 are summarised in Table 4.4.

From Table 4.4, the results are split into several accuracy comparisons:

- i. Comparison of TF-IDF N-grams model accuracy on SVM and CNN
Accuracy on the relevance of each N-grams is analysed whether individually or mix N-grams. Then, the performance of TF-IDF on SVM and CNN is observed.
- ii. Comparison of embedding-based model accuracy on SVM and CNN
Analysing the relevance of each embedding individually or mix embeddings. Then, observe the performance of embeddings (dense vector) on SVM and CNN.
- iii. Comparison between TF-IDF and embedding-based models accuracy on both classifiers. The performance of TF-IDF and embedding-based models are observed based on the combination of embeddings on SVM and CNN.

Table 4.4: Accuracy results of TF-IDF and Embedding-based representation based on the level of feature type

Level	TF-IDF					Embedding-based				
	N-grams	SVM		CNN		Input	SVM		CNN	
		Eng	KD	Eng	KD		Eng	KD	Eng	KD
Word	W1	73.3	66.72	57.66	54	Word	61.18	61.28	96.91	75.06
	W2	61.26	40.54	41.48	43.6					
	W3	43.48	24.98	23.76	38.6					
	W4	27.28	21.4	14.66	39.6					
	W5	18.22	18.4	12.18	34.6					
	W {1:3}	72.84	65.78	60.34	56.2					
	W {1:5}	70.26	64.76	59.98	57.2					
POS	P1	37.36	22.76	67.34	20.4	POS	37.24	37.2	43.43	37.76
	P2	42.52	26.84	63.66	25.2					
	P3	40.38	28.08	31.20	25.6					
Character	C3	72.12	67.7	32.52	58.4	Char	30.90	34.76	56.29	28.37
	C4	70.66	67.94	32.16	58.4					
Mix	W1 + C3	76.40	69.8	62.77	62.4	W+C	61.94	67.76	97.38	90.87
	W1 + C3 + P3	75.12	69.52	68.88	64.4	W+C+P	62.2	68.18	91.5	88.68

i. Analysing of TF-IDF model on SVM and CNN

In earlier studies, combinations of different feature types of N-grams showed a promising result of solving AI issues for online short text (Banga & Mehndiratta, 2017; Rocha et al., 2017). Banga and Mehndiratta (2017) have proven that character 3- and 4-grams perform better yet, they suggested that features combinations yield better results than a single feature. They achieved 75% accuracy using combinations of N-grams (word unigrams, bigrams, character 3-grams, character 4-grams and POS N-grams) represented as TF-IDF where Logistic Regression as the base classifier.

Figure 4.5 illustrates the result of TF-IDF using SVM and CNN. The results in the graph are extracted from Table 4.4. From the table, we simplified the name of features combination, $W1 + C3$ to **Mix 1** and $W1 + C3 + P3$ to **Mix 2** in the graph.

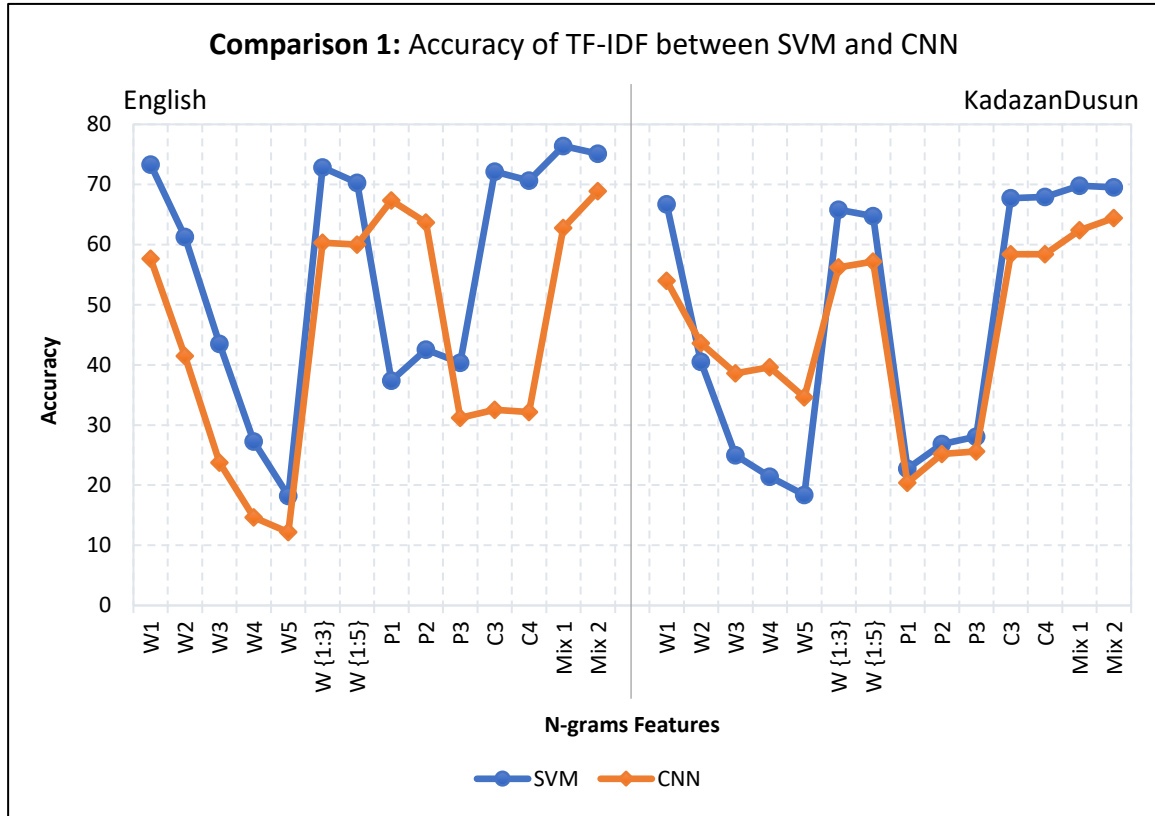


Figure 4.5: Accuracy results of TF-IDF model for both English and KadazanDusun tweets using SVM and CNN as classifiers

Overall, TF-IDF results are in a good agreement with SVM rather than CNN for both corpora (ref. Figure 4.5). The word N-grams features in both corpora; the accuracy dropped sharply over 50% when the value of n increased from 1 to 5. The word N-grams gained back its performance when there are combinations of n in $W \{1:3\}$ and $W \{1:5\}$. The highest value achieved is by word unigrams itself includes outperforming the combinations of n for both corpora.

Moving to POS N-grams, where the POS N-grams on SVM is not more than 45% for English, where the accuracy increase with 5.16% from P1 to P2 but decrease a bit with 2.14% in P3. However, it appears that POS N-grams on CNN recorded a jump in English tweets. P1 recorded highest with 67.34% then starts to fall significantly from P2 to P3 with 32.4% difference. For *KadazanDusun*, the trend from P1 to P3 using both classifiers show progression preceding a bit by SVM. For most cases, POS N-grams seems to have minimal progression by itself with a slow increment from P1 to P3.

Next, character N-grams involves $n = 3$ and 4 to be analysed as individual features. For English, C3 performs better than C4 with a slight difference of 1.46%. Contrary to TF-IDF using CNN, both C3 and C4 show a downward performance with both achieved 32% accuracy. As for *KadazanDusun*, both classifiers act in the same direction but preceded by SVM which led by 9%. Both C3 and C4 features stay at the same level. Among all individual features, C3 shows the best individual performance in terms of accuracy except for English tweets using CNN. C3 and C4 show a very competitive accuracy but, in terms of space reduction, C3 can reduce the n -dimensional matrix yet providing sufficient information which comparable with C4.

Combinations of feature types are done to enhance the AI system accuracy, especially in short social media texts containing very scarce information. There are two types of combination which are, Mix 1 and Mix 2. The increment method of features is done on purpose, as it represents the evolutionary sequence of style features (Zheng et al., 2006). Figure 4.5 reveals that, for corpora, Mix 1 performs better than Mix 2 using SVM with not

Overall, the performance in terms of accuracy of combinations of features in TF-IDF (Mix 1 and Mix 2) provides better performance than individual features. As mentioned

previously, the comparison is on the performance of TF-IDF using SVM and CNN. Evidence shows that most of the results are in good agreement with SVM rather than CNN. It is proven that, CNN is unable to handle well sparse vector represented by TF-IDF with N-grams as the features.

ii. Analysis of Embedding-based Representation Model on SVM and CNN

Embedding is a low-dimensional space (or dense vector) model that captures and preserves the contextual similarity of the input in the embedding space (Developers, 2022). Previous studies prove that embedding is notably successful using deep learning as a classifier (Huang et al., 2020; Khatun et al., 2019; Shrestha, et al., 2017).

Shrestha et al. (2017) proposed a neural network architecture by implementing character embedding to represent the text as a solution to AI for short text. Their best result recorded with 68.3% tested on 1000 tweets of 35 authors. Huang and Iwaihara (2020) presented character and word embedding models using CNN and LSTM as classifiers. They observed that CNN-based models perform better than LSTM with the accuracy of 77.2% on 500 tweets of 50 authors.

It can be seen that most studies paired embedding model with neural network architecture (Theóphilo et al., 2019; Huang et al., 2020; Punchihewa et al., 2021). However, there less study that examined embedding model using classifier other than neural network classifier. Figure 4.6 demonstrates accuracy comparisons between CNN and SVM in handling embedding-based model for both English and *KadazanDusun* corpora. To make the embedding model comparable to TF-IDF, each data level i.e., word, character and POS data, are analysed and their combinations Mix 1 (W + C) outperforms Mix 2 (W + C + P).

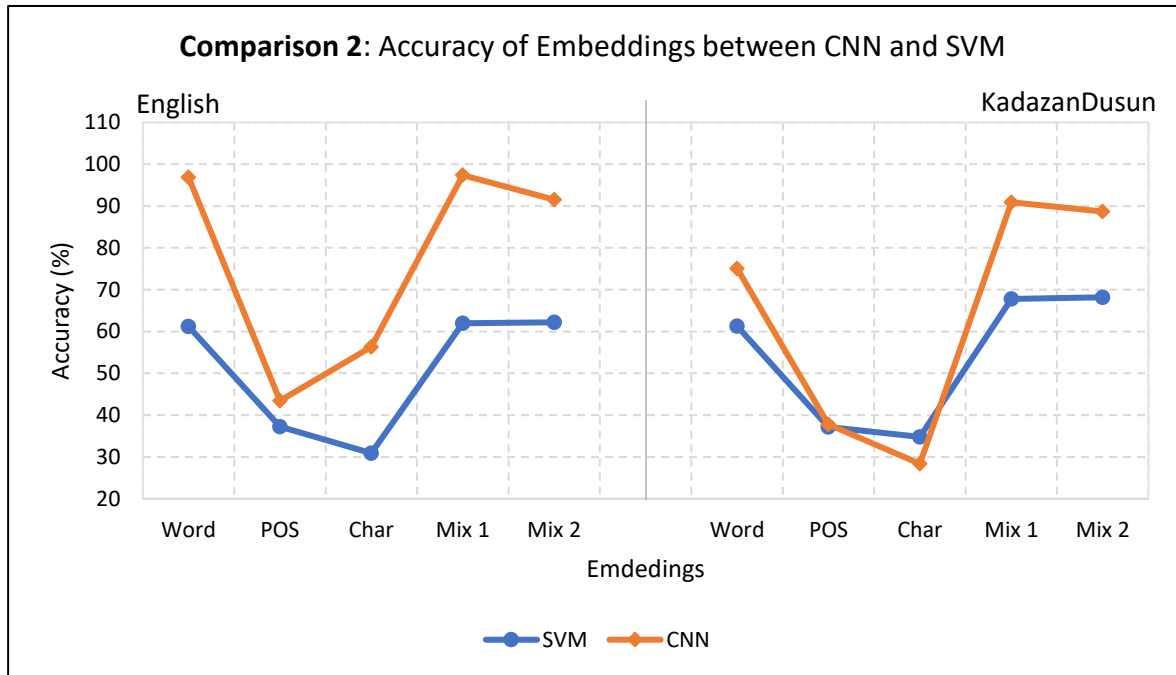


Figure 4.6: Accuracy results of embedding-based model using CNN and SVM for English and KadazanDusun tweets

The embedding-based models, including the individual and combination of inputs for both corpora, show strong concordance on CNN compared to SVM (ref. Figure 4.6). The graph suggests that embedding-based model works fine with CNN but, not with SVM. For English, the performance of the embedding model shoots up to 96.91% for word embedding, leaving POS and character embedding behind with a vast gap. Followed by character embedding with 56.29% accuracy and POS embedding with the lowest accuracy of 43.43%.

The combination of embeddings offers a competitive accuracy using CNN, where Mix 1 outperforms Mix 2 and word embedding with the accuracy of 97.38%. Although Mix 2 carries more information than Mix 1, Mix 2 fail to boost up the. From the result, it is clear that combining POS embedding is not the right decision to do as it will cause a downfall.

As for *KadazanDusun* tweets, the embedding-based model gained the highest accuracy on Mix 1 using CNN with the accuracy of 90.87. Followed closely by the Mix 2 embedding model with 2.19% difference. Word embedding shows the best individual performance in terms of accuracy with 75.06%. The accuracy of character embedding has a considerable drop with only 28.37% accuracy. Similar to English tweets, combining POS embedding with others will cause a slight downfall.

Embedding models on SVM are relatively fair in their performance compared to CNN. For both corpora, Mix 2 model gave satisfactory results from all other embeddings with 62.2% (English) and 68.18% (*KadazanDusun*). Word embeddings show the best individual performance compared to the character and POS embedding. In most cases, Comparison 2 revealed a considerably higher percentage of embedding-based models using CNN compared to SVM for both corpora.

iii. Analysis of TF-IDF Model against Embedding-based Model

Mix 2 is chosen for this analysis to analyse the accuracy comparison between TF-IDF model against the embedding-based model. Mix 2 combinations from N-grams and embeddings are used on SVM and CNN. Mix 2 information (consisting of word, character and POS information) demonstrates reasonably good performance in term of accuracy in previous comparisons with the most information of writing styles. The graph of accuracy is displayed in Figure 4.7.

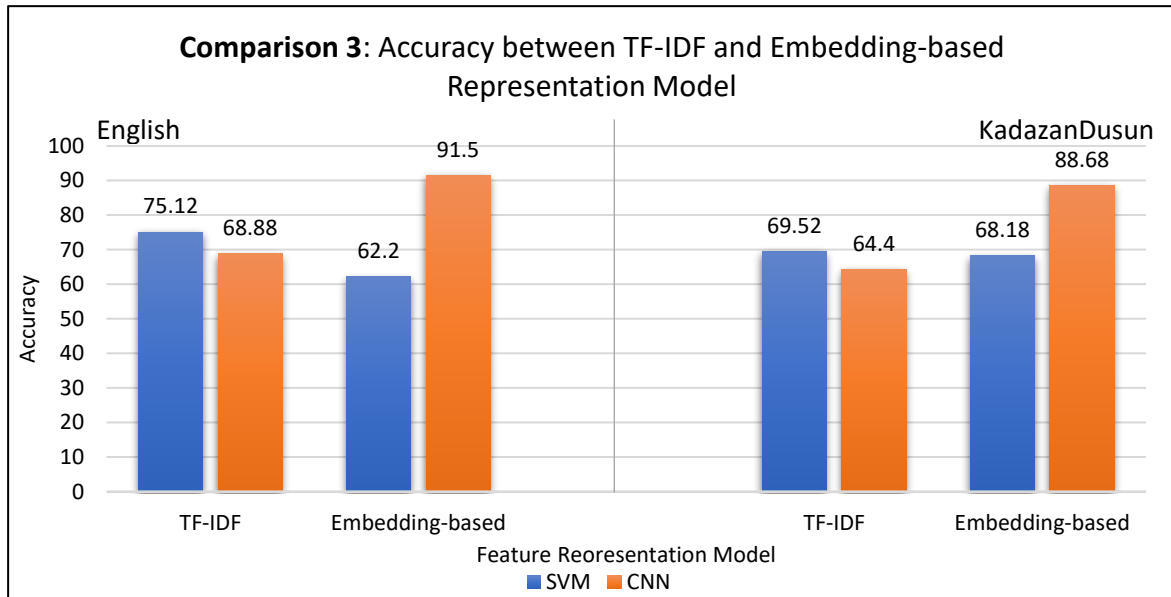


Figure 4.7: Accuracy comparison between TF-IDF and Embedding-based using SVM and CNN as classifiers

The performance of feature representation models in Figure 4.7 indicates the relationship between the models and the classifiers. In both corpora, it can be seen that embedding-based model revealed an excellent result on CNN compared to TF-IDF. The highest accuracy achieved by this model is 91.5% (English) and 88.68% (*KadazanDusun*) using CNN as the classifier. Unfortunately, embedding-based model only excels using CNN but not significant on SVM. The performance plummeted rapidly to 62.2% and 68.18% for English and *KadazanDusun* corpus respectively.

Conversely, TF-IDF gave considerably higher results using SVM. Although the TF-IDF model cannot surpass the significant performance achieved by embedding the model with CNN, the trend shows that the TF-IDF model works better with SVM compared to the embedding model. Observing the corpora, TF-IDF with SVM produces higher accuracy with 75.12% (English) and 69.52% (*KadazanDusun*). There was a gradual decline in embedding model using SVM in English with 12.93% difference.

The graph indicates that embedding model effectively represents short text features if trained with a neural network (in this case is CNN) as the classifier. The reason is that CNN, as a neural network, can work well with dense vector (embedding) rather than sparse vector (TF-IDF). Empirically, it seems that embedding model success in producing excellent performance on short social media text like tweets with given limited source of writing style information; thus, surpassing the TF-IDF model.

It is important to note that classifier also plays a vital role in learning the model besides the representation model. There is a significant correlation between the feature representation model and the classifier to produce a reliable and precise attribution model to predict the author of short social media text. Overall, a dense vector model proves to work well with a neural network; hence, outperforming the other model. In contrast, sparse vector model works well with classifier like SVM. Deduced from this result analysis, TF-IDF model will be paired with machine learning classifiers while embedding-based model will be paired with deep learning classifier in the subsequent analysis for classifiers comparison.

4.2.3 Evaluation of Different Classification Algorithms

In order to evaluate the performance of classification algorithms, the effect of the Independent Variables (IV) on the Dependent Variable (DV) need to be studied (Altakrori et al., 2018). These IVs include the number of candidate authors and the number of tweets per author, which will be examined to find optimum accuracy of the attribution model of AI for SMF. DV is the performance value achieved by the algorithm. The evaluation is done to access the third component of AI system, the attribution method.

4.2.3.1 Setup for Experiment 3

Result analysis from Experiment 2 has shown that TF-IDF with Mix 2 yielded high accuracy. As for embeddings representation, the combination of Mix 2 also achieved the highest accuracy and used to equalise the usage of feature type levels in TF-IDF representation. Deducing from previous analysis, CNN resulted in an outstanding result with embedding representations. Conversely, SVM optimally handles high dimensionality vectors using TF-IDF as the representation model. Figure 4.8 depicts the strategy of Experiment 3.

Experiment 3 is set up by varying the number of tweets and authors accessing the third component of AI to analyse the algorithms' performance. The models are tested by considering 50 authors and a variable number of tweets per user (200, 250, 300, and 500 tweets). The same models were also trained by considering 45 tweets and a variable number of authors (10, 20, 30, 40, and 50 authors).

For TF-IDF model, all randomly sampled sub-datasets are passed to the feature extraction process to extract the features. For embedding model, the data sets are directly converted into embedding vectors as illustrated in Figure 4.8. Later, the representation models are passed to the respective classification algorithms to be trained. CNN trained the embedding while SVM, RF and NB trained the TF-IDF model.

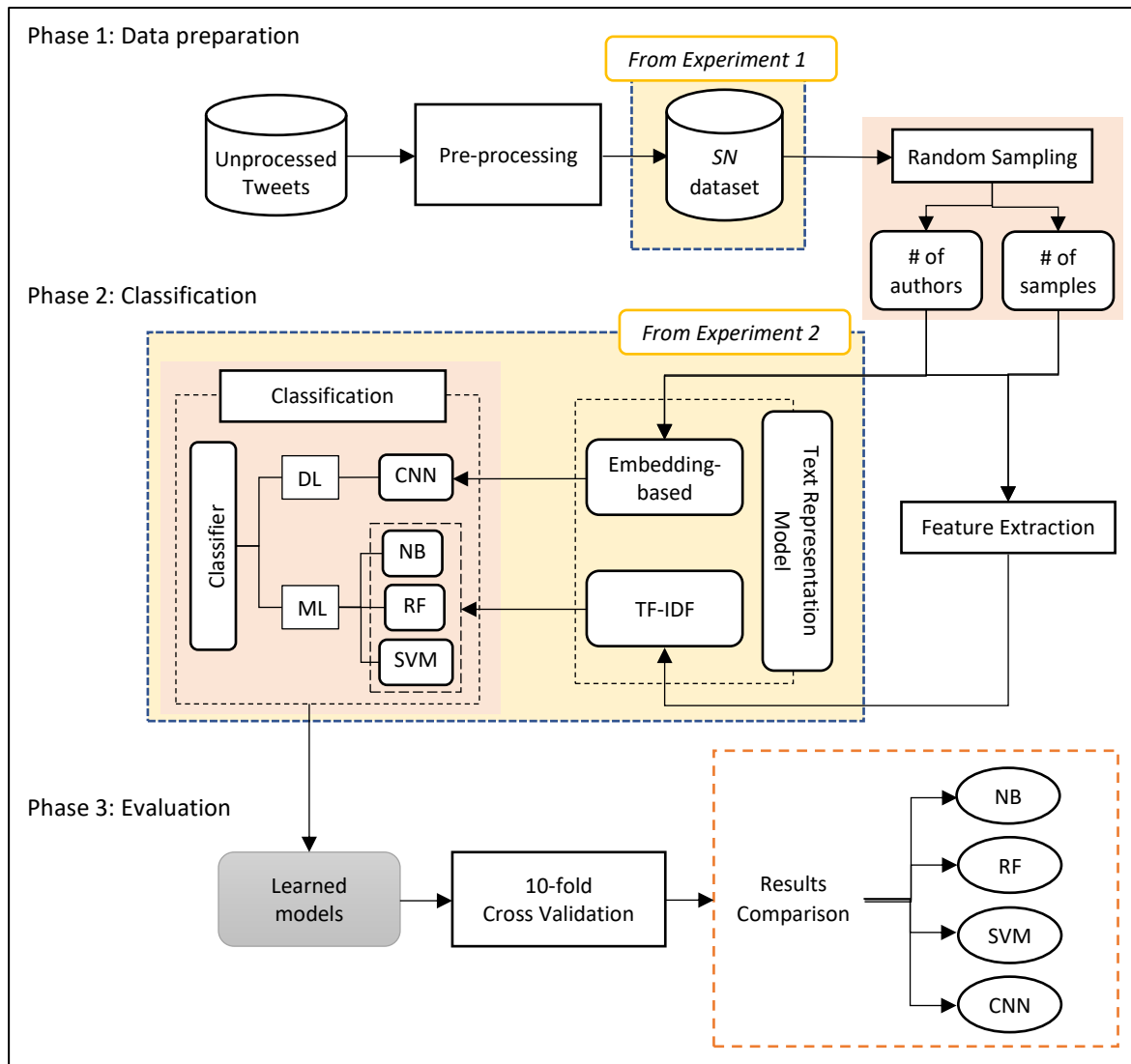


Figure 4.8: Experiment 3 strategy

4.2.3.2 Result Analysis of the Classification Algorithm Performance

To compare the performances of four classification algorithms (NB, RF, SVM and CNN) on AI accuracy given a set of features as mentioned before in the strategy section. Two IVs (number of class and sample) are examined to analyse the DV (performance) whereby, the comparison includes:

- Analysing the performance of classification algorithms (DV) based on a varying number of samples (IV)
- Analysing the performance of classification algorithms (DV) based on a varying number of authors (IV)

i. Analysis of Algorithms' Performances by Varying Number of Samples

Figure 4.9 demonstrates the first performance comparison that includes four different algorithms in varying numbers of tweets. The purpose is to check how much the system can benefit from training on a large corpus.

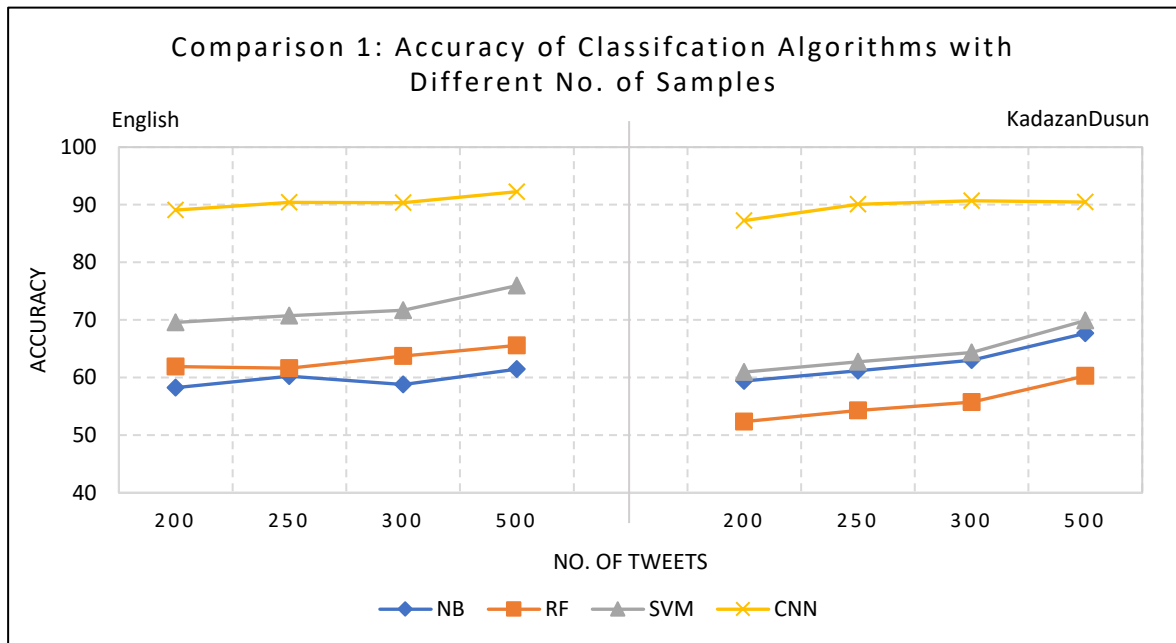


Figure 4.9: Accuracy of four different algorithms with varying number of tweets for English and *KhadazanDusun* tweets

Graph in Figure 4.9 shows that in overall, almost all algorithms in both corpora show an increasing trend with the increase in the number of tweets. The outcome of these four algorithms indicates that the change in the performance (DV) is caused by the increasing

number of tweets per authors (IV). The number of tweets is increased from 250 to 500, but it will not be increased beyond 500 to keep the scenario realistic.

It appears that CNN with embedding model gives an excellent performance for both language corpora outperformed other algorithms. The highest accuracy achieved with 92.27% (English) on 500 tweets and 90.67% (*KadazanDusun*) on 300 tweets. The graph shows that SVM outperformed NB and RF, where TF-IDF as the representation model.

Although unable to beat CNN, adequate performance of SVM in handling TF-IDF also conforms in many previous studies (Mohsen et al., 2016; Schwartz et al., 2013). SVM significantly success in handling short text data, especially social media text, with sparse feature representations. SVM gradually increases as the number of tweets increase and boost up when given 500 samples. Similar behaviour was observed in NB and RF with RF performing better in English, but NB appears to lead.

Table 4.5 displays the precision, recall and f-score of the four algorithms used for classifying *KadazanDusun* tweets. It is apparent that the trends of the scores for both language corpora are in a good agreement; thus, *KadazanDusun* result is chosen for detailed study. In general, the scores have a robust overall concordance with the accuracy results.

Table 4.5: Precision. Recall and F-score of the algorithms for KadazanDusun tweets for different size of samples

No. of samples	NB			RF			SVM			CNN		
	<i>precision</i>	<i>recall</i>	<i>f-score</i>	<i>precision</i>	<i>recall</i>	<i>f-score</i>	<i>precision</i>	<i>recall</i>	<i>f-score</i>	<i>precision</i>	<i>recall</i>	<i>f-score</i>
200	63.08	59.4	58.23	53.88	52.35	51	61.56	60.95	60.6	95	95	95
250	64.95	61.2	60.69	55.84	54.28	52.98	63.26	62.72	62.3	96	96	96
300	65.32	63	62.38	56.68	55.73	54.68	64.8	64.37	64.1	97	97	97
500	71.01	67.98	67.88	62.11	60.28	59.56	70.56	69.92	70	97	97	97

Given more samples of data, the precision score for each algorithm manages to increase the model's sensitivity. In Table 4.5 it can be seen that the measure of predicted actual authors by the model increases as the number of tweets increases. Three algorithms (NB, RF and SVM) indicate that the precision scores are significantly better only when the number of samples is 500. The precision score of CNN remains unchanged when the number of samples increased from 300 to 500.

In a multi-class case, recall is calculated as the sum of true positives across all classes divided by the sum of true positives and false negatives across all classes. Recall scores of NB, RF and SVM are gradually increasing in accordance with the increasing number of samples, which is a good sign to the model. Unlike CNN, the score went up from 200 to 300 tweets but levelled off from 300 to 500 samples.

The F1-score measure is the harmonic mean of precision and recall. It can be seen that the F-score values of the three algorithms (NB, RF and SVM) went up resembling the increase of precision and recall, as the number of tweets increase. CNN shows the same trend but remains stable, although there is an increase in the number of tweets. A good F-score would be indicative of good precision and recall value. CNN outperforms other algorithms in terms of all value followed by SVM with 27% difference for F-score value using 500 tweets sample.

Banga & Mehndiratta (2017) revealed that SVM model proposed by them achieving the accuracy of 73% using a combination of features (char, word and POS n-grams) using 300 samples. The accuracy dropped significantly (67%) when 500 of samples were fed into the classifier. The used of RBF kernel rather than linear in configuring the SVM model could implicate the downfall of the system performance. Similar observation made in Experiment

3, whereby all models suggests that gathering more tweets for training data leads to better performance.

Short text like tweets with only 280 characters provides poor guessing to classifier because they can be immensely affected by random noise. From the experiment, the result revealed that the authors of a very social media text (tweets) could be identified as few as 200 tweets per author by maintaining the accuracy and F-score above 50% for traditional classifiers using TF-IDF in both language corpora. While, While, CNN model can produce relatively high accuracy and maintain optimum F-score with above 90% value although fewer samples are given.

ii. Analysis of Algorithms' Performances by Varying Number of Authors

Figure 4.10 illustrates the findings of the four algorithms' performances for both English and *KadazanDusun* tweets. In this experiment, the tweets were sampled to 45 tweets per author and on data sets containing 10, 20, 30, 40, and 50 authors.

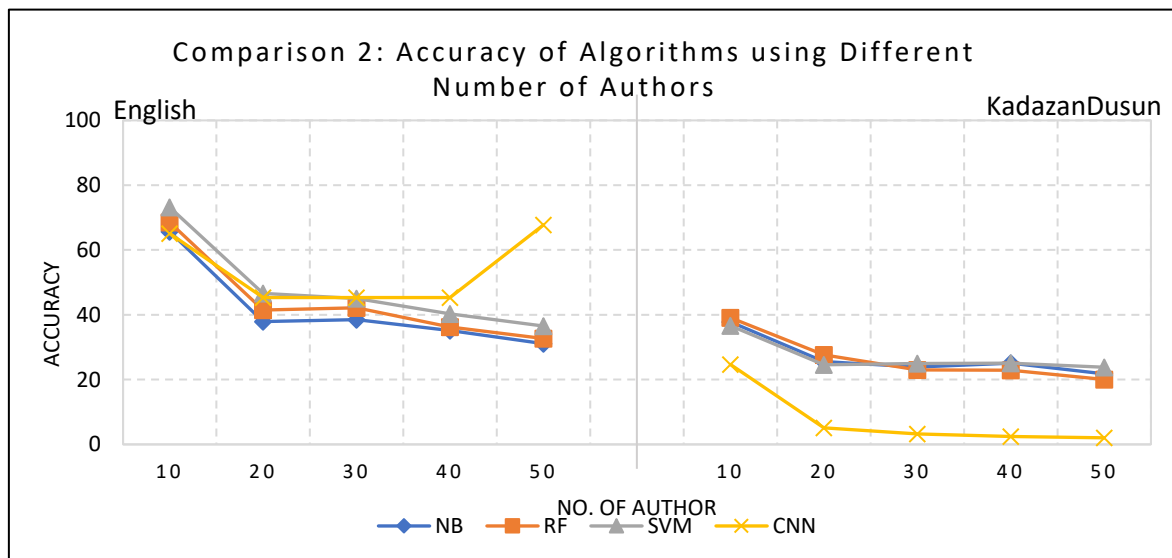


Figure 4.10: The accuracy results of four algorithms with varying number of authors for English and *KadazanDusun* tweets

Graph in Figure 4.10 depicts all four algorithms' accuracy dropped substantially as the number of authors increased. Overall, the results reveal that the second IV (increasing the number of authors) had significant effects on system performance (DV). Using a small number of samples (45 tweets) on ten authors, SVM starts with a good result with 73% accuracy followed by RF, NB and CNN for English. As for KadazanDusun, RF outperforms the rest with 39% accuracy, followed closely by SVM while NB with CNN performance left far behind.

Table 4.6 shows an in-depth finding on the performance of the algorithm by presenting the precision, recall and F-score values. For analysis, *KadazanDusun* results are displayed as both corpora produced similar trend.

Table 4.6: Precision, recall and F-score of four classification algorithms for *KadazanDusun* tweets

No. of samples	NB			RF			SVM			CNN		
	<i>precision</i>	<i>recall</i>	<i>f-score</i>	<i>precision</i>	<i>recall</i>	<i>f-score</i>	<i>precision</i>	<i>recall</i>	<i>f-score</i>	<i>precision</i>	<i>recall</i>	<i>f-score</i>
10	41.07	37.78	35.13	41.09	39.11	37.46	41.12	36.67	35.7	26	33	25
20	24.8	25.56	20.97	27.18	27.67	25.42	22.34	24.56	21.7	0	5	1
30	26.18	23.93	21.21	25.71	22.96	22.06	27.52	24.96	23.9	0	3	0
40	28.98	25.11	24.37	24.26	22.94	21.82	28.98	25.11	24.4	0	3	0
50	23.13	21.78	19.06	21.17	20.04	18.07	26.41	23.82	22.3	0	2	0

Generally, the precision, recall, and F-score results in Table 4.6 of all algorithms are in good agreements with accuracy. The results indicate a significant drop when the number of authors was increased from 10 to 50 authors, especially starting from 20 authors and above. It appears that the precision, recall and F-score value of CNN were somewhat impossible to provide any value after the number of authors is increased to 20 and above. It is difficult for CNN with embedding model to learn the information with very few data (only

45 tweets) with a high number of classes. Usually, neural network classifier like CNN naturally required an abundance of information to pre-train the model.

Altakrori et al. (2018) proved that from the test they conducted, the performance of algorithms with an increasing number of authors significantly decreases in terms of the accuracy. RF recorded with the highest accuracy of 58% outperformed other classifiers including SVM and NB, given 45 tweets per 10 authors to be classified. As the number of authors is increased to 20, the algorithms decreased gradually. They suggested a larger number of additional tweets are needed to have a statistically significant increase in the model's performance.

The results in both Table 4.6 and Figure 4.10 revealed that the performance of the algorithm decreases drastically when the number of authors increases. As the number of classes (authors) increases, it is possible that fewer data present to specify and differentiate each class for the classification algorithm. Apart from that, a high dimension of vectors means a very large space that complicates the algorithm to search with a large number of classes given only few training samples.

However, it should be noted that CNN model fails to maintain the performance with few samples of data given and more authors are needed to be classified. Although SVM, RF and NB were unable to beat CNN in previous comparisons, it seems that given a situation of very few samples given TF-IDF representation can be learned remarkably by these algorithms. In spite, classifier with TF-IDF model remains relevant in maintaining the performance with the increasing number of authors.

4.3 Discussion

In this section, the implications of each evaluation towards SMF are discussed. Suggestions in increasing AI performance specifically in the SMF field are provided based on the observation in result analyses before.

4.3.1 Utilising Twitter Native Features in Improving the Performance of AI

The main goal of Experiment 1 is to answer the RO 1 of which is to analyse the importance of Twitter native features in AI by comparing the presence and absence of them in the input data (tweets). A study done by Kebede et al. (2015) reveals that the use of hashtags in detecting the similarity of the author's writing style in tweets increase by 8% from 53%.

Result analysis in Experiment 1 confirmed that the presence of native features of Twitter proves to enhance the AI system in attributing the author of short social media text. Both corpora show an increment of accuracy when the input dataset contains native features of Twitter in it. Banga & Mehndiratta (2017) chose to remove the information regarding Twitter in tweets including URLs, user references and punctuations with their best results of 71.25% accuracy. While, by including the native features, the AI performance can boost up to 90% (see Figure 4.3).

Findings in Experiment 1 provide compelling evidence that besides stylometry, the native features of social media application provide support by providing sufficient information to identify the author of short social media text. Interestingly, the native features unlikely fickle or affected by the changes of topics and genre in the tweets. Normalising the native features into standard tags helps to generalise the information and make them not specific to certain topics or genres. Based on the observation in Experiment 1, including the

native features not only can improve the accuracy of AI, besides these features could help overcome the issue of diversity of genres and topic in AI application for social media text.

An important question for further studies is to determine other elements that native to Twitter and exploit them to hone the performance of AI, especially for social media text. It should be possible to include the native features of Twitter onto other techniques in AI to facilitate in revealing the criminal that perpetrates cyberbully activities on Twitter.

4.3.2 Dense Representation for Better Performance in AI for SMF

The primary purpose of conducting Experiment 2 is to answer the RO 2 of which feature representation could best represent the authors' writing style in AI system for SMF? Experiment 2 has demonstrated two feature representation model, namely, TF-IDF N-grams and embedding-based model. TF-IDF N-grams model is represented as a sparse vector while embedding-based model represented as dense vector.

Previous studies acknowledged character n-grams as the most accurate style marker in solving short text issue in AI achieving 80% precision using more than 1000 tweets (Azarbonyad Te al., 2015). Later, researchers modified the model by combining with other features such as word N-grams (representing lexical features) (Gómez-Adorno et al., 2017) with average F-score of 57.3% and POS N-grams (representing syntactical features) (Banga & Mehndiratta, 2017) with improvement by 71.25% using 200 tweets. Experiment 2 presented TF-IDF models with the presence of Twitter native features, 76.40% with 500 tweets.

Problem with N-grams is that they generate very sparse vectors of high dimensionality. To add, TF-IDF provides the text representations without clear semantic

interpretation. Alternatively, embedding-based model offers a solution to overcome sparsity issues by representing text in a dense vector. Embedding model hones the performance of AI with an accuracy of 97.38% using CNN as the classifier.

With only 300-dimension vector, this model significantly reduces the features yet preserving the contextual similarity of the text in the corpus. The results provide a clear distinction between sparse and dense model in representing short text data for classification. Experiment 2 provides insight into the embedding-based model, which remarkably improves the performance of the AI system by representing short text in a denser form with a relatively low dimension.

However, the advantage of embedding model over TF-IDF lies in combining the classification algorithm to learn the model. Embedding model exceptionally provides high accuracy when using a neural network as the classification algorithm, yet, it did not perform well with SVM. It can be speculated that the representation of data in embedding model, i.e., word, character, and POS input, is insufficient to generalise the model and discriminate well the author's writing style. The implication of such a situation may affect the performance of the learned model.

Embedding-based model with input, i.e., word, character and POS have shown a very remarkable result in enhancing the performance of AI, but sequence-based input (e.g., a sequence of characters) may be considered in the future to investigate the effect of using this type of input in representing text in embedding model.

From the finding, the embedding model can well-discriminate the writing styles by capturing meanings of words, semantic relationships and different types of content in tweets. N-grams create an increase in the number of dimensions. Hence, it would be beneficial to

enact feature selection techniques in the future to select important features and reduce the sparsity and for better representation before being learned by the classifier.

4.3.3 Maintaining Optimal Performance of Classification Algorithm in AI for SMF

The main purpose of Experiment 3 is to answer the RO 3 regarding which classification algorithm able to maintain the accuracy of AI system. In Experiment 3, it could be seen that these algorithms are evaluated based on a varying number of authors and samples to examine their performance. Prior work has documented that the typical advantage of SVM achieving unexceptional well performance in learning the attribution model to predict the author of the text (Fourkioti et al., 2019; Jambi et al., 2021).

However, Experiment 3 shown that SVM unable to maintain accuracy if less data is given to be classified, whereby only 45 tweets were prepared. Interestingly, other algorithms seem to perform similarly with SVM, which in this case, somehow refute the theory of SVM performed well in AI. Yes, it does perform only if SVM is given a sufficient amount of data to be trained. In the case of forensics scenario, to obtained abundance of evidence for a criminal is very hard and rare. Besides, the performance of SVM decreases slightly when being tested on *KadazanDusun* corpora, although a huge number of training samples.

The main reason for lower performance in *KadazanDusun* might be probably related to the preparation of data. During preparation, much stylistic information is removed, i.e., words other than native words. Besides, tagging in *KadazanDusun* is done using standard POS tagger for English which can affect the performance of the classifier. Another issue might be related due to imbalance data represented by each author in forms of word distribution. Although the number of samples is balanced, the number of words in each tweet is not the same.

The imbalance number of words creates an imbalanced representation of writing style (words) for each author. It seems that this is the limitation of implementing an instance-based approach for short text problem. Future work should explore the fusion of instance-based and profile-based approach. This approach works by representing all text samples differently (instance-based), but the representation vectors for the text for each author are averaged in feature-wise form producing a single profile vector for each author (profile-based).

Experiment 3 is deduced from the findings of Experiment 2 and 1. The input data contain the native feature of Twitter as additional information stylometry is used as the data set. The representation of text for CNN is different compared to the other three algorithms. CNN utilised embedding model, which indirectly represent a deep learning model in this experiment. Result in Experiment 2 provides a clear distinction between TF-IDF model and embedding model pairing with CNN, so the embedding model is being applied to CNN. In a forensic case, it is critical for an attribution model is reliable and precise in attributing the evidence (text) to the original writer of the text (criminal).

Thus, choosing the right model by pairing the representation model and classifier is vital in building a reliable attribution model. According to Chowdhury et al. (2019), neural networks do not play well with high-dimensional data. Embeddings are dense representations of words or characters that capture the meaning and context of the text. CNN is well-suited for AI task by capturing localized patterns in the text, as authors tends to have their own unique writing style that is reflected in the local pattern of their text. Sequence models in other hand is not suitable for AI task because the models are trained to recognize pattern across time (long-range dependencies). Sequential models capture sequential

information from the text to predict the next word where AI task is not sequential task and does not require the order of the words to be preserved hence, making CNN is more suitable for AI task (Chowdury et al., 2019).

CNN takes advantage of low-dimensional data by capturing a similar pattern provided by the features. The outcome of each convolution filters will be stimulated when a unique pattern is detected. In AI task, However, the downside of CNN is that it is unable to produce a generalised model that conform a few training samples. It requires practitioners to tune the hypermeters regards to the exact model that is being pre-trained. In the case of very low training dataset provided (45 tweets), the hypermeters setting is tuned based on the pre-trained model with a large number of training samples. Thus, Experiment 3 provides insight for future studies to access the performance of CNN in identifying the author of very short and limited text such as tweets by exploring the hyper-parameters.

4.4 Summary

Chapter 4 present the evaluation done to the proposed AI system for social media forensic using tweets as the test corpus. Two test corpora are utilised to evaluate the system which are, English and *KadazanDusun*. The purpose of using two corpora is to compare the proposed system, whether it can handle other language data than English. *KadazanDusun* language is chosen as it represented the indigenous language in Malaysia with an intense number of native speakers across Sabah.

The evaluation on the proposed system is broken into three main components of AI composing of the input or data, features, and the attribution method. Three experiments are being conducted which designed to answer the research questions of the study. All the experiments are introduced in different subsections, starting with explaining the

experiments' strategy; then, the results analysis is presented and in-depth analysis of the result is done based on the graphs.

The finding shows that Twitter native feature is essential in enhancing the accuracy for AI system, which deals with short, limited text like tweets. Dense representation seems to adapt better with CNN compares to SVM and befitting the AI system producing remarkable high accuracy. The performance of algorithms shows a pitfall when a limited amount of data is fed for training with a high number of authors. CNN was unable to perform well in attributing *KadazanDusun* tweets due to over cleaning the data during the data preparation process. Besides, there are insufficient data to pre-trained the embedding model for *KadazanDusun*.

The evaluation has shown extensive comparisons of the major components in AI system and discusses their implications in AI towards SMF field. The mind map of Chapter 4 is included in Appendix C. Next chapter will cover the conclusion of the research.

CHAPTER 5

CONCLUSION

5.1 Conclusion

AI is pertinent in assisting digital forensics investigation to track down the cyberbully perpetrators in social media. Advance network technology such as, Tor, provides anonymity service which complicates current digital forensics to trace the perpetrators online. Text and messages posted online are only clues left to trace the real culprit, which is achievable through AI. AI is a task where the candidate authors' writing samples are analysed to identify the most plausible author of the anonymous text. Hence, a proposed workflow of AI for SMF is presented. Tweets corpora of different languages, English and an Under-resourced language (*KadazanDusun*) are used to evaluate the proposed workflow. OSN messages like tweets are chosen because tweets are best represented as short online messages freely accessible for developers and researchers.

This research aims to tackle the issue of short text in AI in the context of SMF. The main gap of AI is maintaining its performance in handling short text data like OSN messages. Limited information in short OSN messages like tweets, harden AI task in analysing the authors' writing styles and accurately predicting the author of anonymous text. The proposed workflow is focused on enhancing the performance of AI by accessing the major components of AI to propose the optimum AI solution in handling short OSN text for SMF. In-depth analyses are done on the proposed workflow based on the major components of AI (refer to Chapter 4). There are three types of evaluations on the proposed workflow: evaluation on the input data, evaluation on the text representation models, and evaluation of the classification algorithms.

This chapter gives an overview of the work done in this thesis by aligning it with the research objectives (RO_n) and research contributions (RC_n). The limitations of the current work will also be discussed. Possible further improvements on the current work will be discussed in the future works.

5.1.1 Research Contributions

The following Figure 5.16 depicts the mapping of the ROs to the RCs of the work done in this research as in Chapter 1.

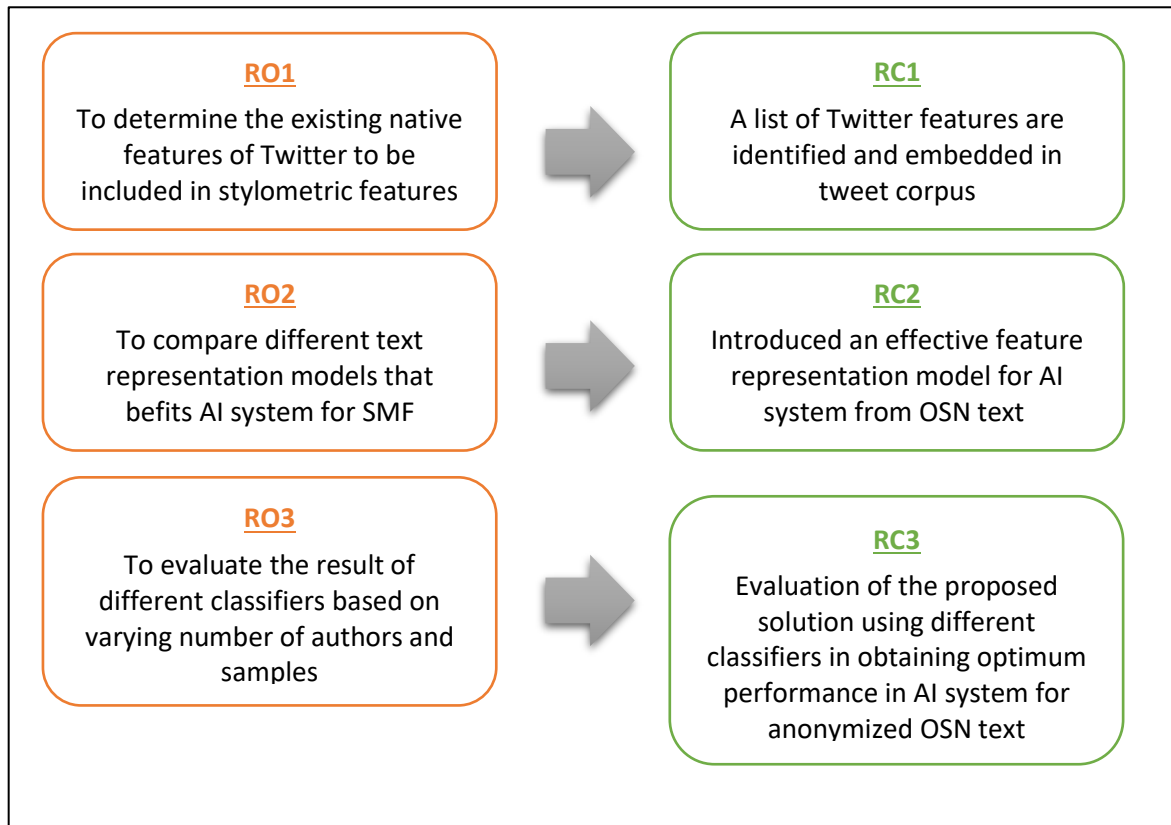


Figure 5.16: Mapping of ROs to RCs

Based on Figure 5.16, the followings subsections are the elaborations of the RCs.

i. The Creation of Tweets Corpus with the Presence of Twitter Native Features

The first contribution includes the creation of a new corpus for AI, especially in SMF. The corpus consists of a collection of tweets with the inclusion of Twitter native features. The inclusion of Twitter native features in tweets has proven to enhance the performance of the AI system in SMF through Experiment 1 evaluation (refer Chapter 4). The nature of OSN text like tweets is different from the formal text in terms of the short text and the irregular language.

Therefore, to close up the gap of short text issue poses by OSN text in AI is by availing the native elements of the OSN text. Twitter native features, i.e., URL links, user references and hashtags, could provide salient information in representing the writing style of social media user disregards specific genre and topics of the text.

Besides English tweets, there is a corpus building for under-resourced language (U-RL), KadazanDusun tweets. The creation of U-RL tweet corpus can help promote the research in fields such as natural language processing and text analysis, particularly in SMF area to combat cyberbullying involving native languages.

ii. Discovery of Feature Representation Model Befitting the AI System from OSN text

It is crucial to explore the finest possible way in representing limited information in tweets so that the information could be represented to the maximum. N-grams features have proven to work best in representing the author's writing style for short text. Using the TF-IDF text representation model to represent the N-grams, the dimension of the matrices expands as more text is fed in to be trained.

This research divulges an alternative method to represent the tweets in a lesser sparse way: the embedding-based model. The embedding models evinced higher accuracy than TF-IDF N-grams in Experiment 2 evaluation (refer Chapter 4).

The findings also suggested that embedding models work best with neural network algorithm as the classifier using the deep learning approach and not with traditional machine learning algorithm, i.e., SVM. Neural network learns well on dense vector represented by the embedding model while traditional classifier like SVM could not learn effectively with such representation.

iii. Assessments of Different Classifiers on the Proposed Solution of AI for OSN text

Studies have shown that SVM maintains the best accuracy in classifying short text in AI using Bag-of-Word approach model, i.e., TF-IDF (Fourkioti, et al. 2019; Jambi, et al., 2021). Recently, a robust neural network classifier in image processing, Convolutional Neural Network (CNN), shown an excellent performance in AI field by deploying the embedding model to represent text (Theóphilo et al., 2019; Barlas & Stamatatos, 2021).

Naïve Bayes and Random Forest are being employed to compare the performances with the state-of-the-art AI models. Naïve Bayes is well known for producing simple naïve assumption model in classifying text. Meanwhile, Random Forest is chosen as this classifier capable of finding the most important features in classification besides being an ensemble model by aggregating the result produced by multiple decision trees.

This research exhibits various AI models using four different classifiers. Varying number of authors and tweets were used to assess the classifiers' performance. The assessment on the classifiers' performance is vital to find the most optimum solution that can

maintain the performance of AI system in handling peculiar situation during forensics investigation. The findings obtained in the third evaluation on the classifiers suggest that an adequate quantity of the author's writing samples is needed to maintain the optimal performance of the classifier.

The findings also recommended that too many candidate authors present during classification will reduce the performance of the classifier. Assessment on the model performance is essential to validate the state-of-the-art AI model in SMF area due to the increasing number of OSN users and messages in online communication, which can affect the accuracy of AI system.

5.2 Limitations

5.2.1 The Inadequacy of Language Resources for U-RL

Evaluations on the proposed workflow of AI using U-RL corpus resulted in lower accuracy compared to English. The main factor affecting the model's performance is the lack of language resources (i.e., dictionary thesaurus) and language independent NLP tools that specifically handle the U-RL. Besides that, insufficient textual information in U-RL tweets is one of the contributions that lessen the AI model's performance.

5.2.2 Imbalance of Word Distribution in Tweets

One of the biggest challenges in deploying instance-based approach to represent the author writing style is the word distribution imbalance problem. Although powerful classification algorithms are used to build a predictive model of AI, the text-length of the training text sample plays a vital role to produce a model with optimal performance. Uneven word distribution over the candidate authors greatly impacted the predictive model by misclassifying the tweets and resulted in the inaccurate prediction made by the model.

5.2.3 Model Sparsity

N-grams model introduced in the proposed workflow consisted of dynamic features extracted from tweets containing few words, thus producing a sparse feature vector. All tokens, including punctuations, are included without removing any of them with the hypothesis of more data represented the more accurate the predictive model. However, due to the overwhelming number of distinct features produce by N-grams model, it failed to converge the writing style. In this case, skipping the feature selection process might lead to worse performance in terms of accuracy and speed.

5.3 Future Works

5.3.1 Conventional Approach in Processing U-RL Data

U-RL corpus, which contains a collection of U-RL language tweets, is crucial to enable AI system analysed U-RL text which prone to cyberbullying to facilitate forensics investigation. However, due to limited language resources, the approach used to process the U-RL text is unseemly suitable. Using the standard NLP tools to pre-process the U-RL text is inappropriate, which resulted in lower accuracy of the AI system. Thus, in future, it is crucial to conduct a proper approach in handling and processing the U-RL text, for instance, utilising unsupervised POS tagger that able to POS tagged the U-RL text. The recommended solution could help hone the AI system's performance in handling U-RL text in the future.

5.3.2 Hybrid-based Approach to Represent the Author's Writing Style

Due to the complexity of the short text issue in AI for SMF, combining different approach, i.e., such as fusing profile-based and instance-based approaches, seems to be a promising path. In the future, the hybrid approach can be implemented by borrowing some elements from both approaches. For example, all training samples are represented separately,

and the representation vectors for the samples are averaged, producing a single vector of text representation for each author. Consequently, the hybrid approach can overcome the data imbalance problem to represent tweets over the candidate authors. Mind map of Chapter 5 is presented in Appendix C.

REFERENCES

- Abbasi, A., & Chen, H. (2005). Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems*, 67-75.
- Abbasi, A., Javed, A. R., Iqbal, F., Jalil, Z., Gadekallu, T. R., & Kryvinska, N. (2022). Authorship identification using ensemble learning. *Scientific Reports*, 1-16.
- Abdullah Al-Ajlan, M., & Ykhlef, M. (2018). Deep learning algorithm for cyberbullying detection. *International Journal of Advanced Computer Science and Applications*, 199-205.
- Altakrori, H., M., Iqbal, F., Fung, B. C., Ding, S. H., & Tubaishat, A. (2018). Arabic authorship attribution: An extensive study on twitter posts. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 1-51.
- Apin, P., & Wahab, K. A. (2015). Tabu bahasa dalam masyarakat Dusun di Daerah Ranau, Sabah. *Jurnal Melayu*, 224-239.
- Argamon, S., Koppel, M., Pennebaker, J. W., & Schler, J. (2009). Automatically profiling the author of an anonymous text. *Communications of the ACM*, 52(2), 119-123.
- Ariffin, A., Mohd, N., & Rokanatnam, T. (2021). Cyberbullying via Social Media: Case Studies in Malaysia. *OIC-CERT Journal of Cyber Security*, 21-30.
- Azarbonyad, H., Dehghani, M., Marx, M., & Kamps, J. (2015). Time-Aware Authorship Attribution for Short Text Streams. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 727-730). Santiago: ACM.

- Baig, A. R., & Kayani, H. M. (2015). Swarm intelligence based author identification for digital typewritten text. *In 2015 First International Conference on Anti-Cybercrime (ICACC)* (pp. 1-6). Riyadh: IEEE.
- Balakrishnan, V., Khan, S., & Arabnia, H. R. (2020). Improving cyberbullying detection using Twitter users' psychological features and machine learning. *Computers & Security*, 101710.
- Banga, R., & Mehndiratta, P. (2017). Authorship attribution for textual data on online social networks. *2017 Tenth International Conference on Contemporary Computing (IC3)*. Noida: IEEE.
- Barlas, G., & Stamatatos, E. (2021). A transfer learning approach to cross-domain authorship attribution. *Evolving Systems*, 625-643.
- Bhargava, M., Mehndiratta, P., & Asawa, K. (2013). Stylometric analysis for authorship attribution on twitter. *In International Conference on Big Data Analytics* (pp. 37-47). Cham: Springer.
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 135–146.
- bullyingstatistic. (7 July, 2015). *Cyber Bullying Statistics*. Retrieved from Bullying Statistics: <http://www.bullyingstatistics.org/content/cyber-bullying-statistics.html>
- Carvalho, V. F., Nascimento, C., & Nogueira, B. M. (2020). Machine Learning for Suicidal Ideation Identification on Twitter for the Portuguese Language. *In Brazilian Conference on Intelligent Systems* (pp. 536-550). Cham: Springer.

- Chaski, C. (2005). Who's at the keyboard? Authorship attribution in digital evidence investigations. *International Journal of Digital Evidence*, 1-13.
- Chaski, C. (2007). The keyboard dilemma and authorship identification. In *IFIP International Conference on Digital Forensics* (pp. 133-146). New York: Springer.
- Cheng, L., Guo, R., Silva, Y., Hall, D., & Liu, H. (2019). Hierarchical attention networks for cyberbullying detection on the instagram social network. In *Proceedings of the 2019 SIAM International Conference on Data Mining* (pp. 235-243). Calgary: Society for Industrial and Applied Mathematics.
- Chowdhury, H. A., Imon, M. A., Hasnayeem, S. M., & Islam, M. S. (2019). Authorship Attribution in Bengali Literature using Convolutional Neural Networks with fastText's word embedding model. *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)* (pp. 1-5). Dhaka: IEEE.
- Corney, M. W., Anderson, A. M., Mohay, G. M., & Vel, O. d. (2001). Identifying the authors of suspect email. *Communications of the ACM*.
- De Vel, O., Anderson, A., Corney, M., & Mohay, G. (2001). Mining e-mail content for author identification forensics. *ACM Sigmod Record*, 55-64.
- Developers. (18 July, 2022). *Embeddings*. Retrieved from Developers: <https://developers.google.com/machine-learning/crash-course/embeddings/video-lecture>
- Dharma, E. M., Gaol, F. L., Leslie, H., Warnars, H. S., & Soewito, B. (2022). The accuracy comparison among word2vec, glove, and fasttext towards convolution neural

- network (cnn) text classification. *Journal of Theoretical and Applied Information Technology*, 31.
- Diurdeva, P., Mikhailova, E., & Shalymov, D. (2016). Writer identification based on letter frequency distribution. In *2016 19th Conference of Open Innovations Association (FRUCT)* (pp. 24-30). 2016: IEEE.
- Domo. (2021). *DOMO*. Retrieved from Data Never Sleeps 9.0: <https://www.domo.com/learn/infographic/data-never-sleeps-9>
- Doultani, M. A., & Vijayalakshmi, M. (2019). Data Forensics On Social Media. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)* (pp. 1-5). Bombay: IEEE.
- Eder, M. (2015). Does size matter? Authorship attribution, small samples, big problem. *Digital Scholarship in the Humanities*, 30(2), 167-182.
- Eyuboglu, M., Eyuboglu, D., Pala, S. C., Oktar, D., Demirtas, Z., Arslantas, D., & Unsal, A. (2021). Traditional school bullying and cyberbullying: Prevalence, the effect on mental health problems and self-harm behavior. *Psychiatry Research*, 113730.
- Ferreira, M. J. (2017). *Workflow Recommendation for Text Classification Problems*. Porto: University Do Porto.
- Fourkioti, O., Symeonidis, S., & Arampatzis, A. (2019). Language Models and Fusion for Authorship Attribution. *Information Processing & Management*, 102061.
- Frommholz, I., al-Khateeb, H. M., Pothast, M., Ghasem, Z., Shukla, M., & Short, E. (2016). On Textual Analysis and Machine Learning for Cyberstalking Detection. *Datenbank Spektrum*, 16(2), 127–135.

- Gladwin, A. A., Lavin, M. J., & Look, D. M. (2017). Stylometry and collaborative authorship: Eddy, lovecraft, and ‘The Loved Dead’. *Digital Scholarship in the Humanities*, 32(1), 123-140.
- Gómez-Adorno, H., Alemán, Y., Vilariño, D., Sanchez-Perez, M. A., Pinto, D., & Sidorov, G. (2017). Author clustering using hierarchical Clustering analysis: Notebook for PAN at CLEF 2017. In *CEUR Workshop Proceedings (Vol. 1866)* (pp. 1-7). CEUR-WS.
- Goni, O., Ali, M. H., Showrov, Alam, M. M., & Shameem, M. A. (2022). The Basic Concept of Cyber Crime. *Journal of Technology Innovations and Energy*, 16-24.
- Green, R. M., & Sheppard, J. W. (2013). Comparing Frequency- and Style-Based Features for Twitter Author Identification. *Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference* (pp. 64-69). Association for the Advancement of Artificial.
- Hamad, N., & Eleyan, D. (2022). Digital Forensics Tools Used in Cybercrime Investigation – Comparative Analysis. *Journal of Xi'an University of Architecture & Technology*, 113-126.
- Han, Z., Wang, Z., & Li, Y. (2021). Cyberbullying involvement, resilient coping, and loneliness of adolescents during Covid-19 in rural China. *Frontiers in Psychology*, 2275.
- Hewling, M. O. (2009). *Digital forensics: an integrated approach for the investigation of cyber/computer related crimes*. Luton: University of Bedfordshire.
- Hitschler, J., Berg, E. V., & Rehbein, I. (2018). Authorship attribution with convolutional neural networks and POS-eliding. In *Proceedings of the Workshop on Stylistic*

- Variation (EMNLP 2017)* (pp. 53-28). Copenhagen: The Association for Computational Linguistics.
- Hon, L. C., & Varathan, K. D. (2015). Cyberbullying Detection System on Twitter. *In International Journal of Information Systems and Engineering (IJISE)*, 1(1), 1-11.
- Howedi, F., & Mohd, M. (2014). Text classification for authorship attribution using Naive Bayes classifier with limited training data. *Computer Engineering and Intelligent Systems*, 5(4), 48-56.
- Huang, W., Su, R., & Iwaihara, M. (2020). Contribution of improved character embedding and latent posting styles to authorship attribution of short texts. *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data* (pp. 261-269). Cham: Springer.
- Intuition Engineering. (25 February, 2019). *Chars2vec: character-based language model for handling real world texts with spelling errors*. Retrieved from Hackernoon: <https://hackernoon.com/chars2vec-character-based-language-model-for-handling-real-world-texts-with-spelling-errors-and-a3e4053a147d>
- Jambi, K. M., Khan, I. H., Siddiqui, M. A., & Alhaj, S. O. (2021). Towards Authorship Attribution in Arabic Short-Microblog Text. *IEEE Access*, 128506-128520.
- Juola, P. (2008). Authorship attribution. *Foundations and Trends in Information Retrieval*, 233-334.
- Kaikini, S., Agarwal, K., Raj, K., & Kulshrestha, S. (2022). Statics on Cyber Crimes and Cyber Laws in India: A Study. *Journal of Algebraic Statistics*, 2935-2944.

- Kebede, A. M., Tefrie, K. G., & Sohn, K.-A. (2015). Anonymous author similarity identification. *In 2015 5th International Conference on IT Convergence and Security (ICITCS)* (pp. 1-5). Kuala Lumpur: IEEE.
- Khanh, B., & Vorobeva, A. (2020). A preliminary performance comparison of machine learning algorithms for web author identification of vietnamese online messages. *2020 26th Conference of Open Innovations Association (FRUCT)* (pp. 166-173). Yaroslavl: IEEE.
- Khatun, A., Rahman, A., Islam, M. S., & Marium-E-Jannat. (2019). Authorship Attribution in Bangla literature using Character-Level CNN. *2019 22nd International Conference on Computer and Information Technology (ICCIT)* (pp. 1-5). Dhaka: IEEE.
- Koppel, M., Argamon, S., & Shimon, A. R. (2002). Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17(4), 401-412.
- Koppel, M., Schler, J., & Argamon, S. (2009). Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology*, 60(1), 9-26.
- Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text Classification Algorithms: A Survey. *Information*, 150.
- Le, Q., & Mikolov, T. (2014). Distributed Representations of Sentences and Documents. *Proceedings of the 31st International Conference on International Conference on Machine Learning* (pp. 1188-1196). Beijing: ACM.
- Liu, D. (2009). Cisco router and switch forensics: Investigating and analyzing malicious network activity. Syngress.

- Loades, M. E., Chatburn, E., Higson-Sweeney, N., Reynolds, S., Shafran, R., Brigden, A., Crawley, E. (2020). Rapid systematic review: the impact of social isolation and loneliness on the mental health of children and adolescents in the context of COVID-19. *Journal of the American Academy of Child & Adolescent Psychiatry*, 1218-1239.
- Mahajan, D., Patil, R., & Sankar, V. (2018). *Word2Vec using Character n-grams*. CA: Stanford.
- Markov, I. J.-L. (2017). Authorship attribution in portuguese using character n-grams. *Acta Polytechnica Hungarica*, 14(3), 59-78.
- Meijer, H. J., Truong, J., & Karimi, R. (2021). Document embedding for scientific articles: Efficacy of word embeddings vs TFIDF. *arXiv preprint arXiv:2107.05151*, 1-12.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv:1301.3781*, 1-12.
- Montasari, R., Hill, R., Carpenter, V., & Montaseri, F. (2019). Digital forensic investigation of social media, acquisition and analysis of digital evidence. *International Journal of Strategic Engineering (IJoSE)*, 52-60.
- Moore, M. J., Nakano, T., Enomoto, A., & Suda, T. (2012). Anonymity and roles associated with aggressive posts in an online forum. *Computers in Human Behavior*, 861-867.
- Mulazzani, M. (2014). *New challenges in digital forensics: online storage and anonymous communication*. [Doctoral dissertation, Vienna University of Technology]. Retrieved from <https://repositum.tuwien.at/handle/20.500.12708/5835>
- Ngejane, C., Eloff, J., Sefara, T., & Marivate, V. (2021). Digital forensics supported by machine learning for the detection of online sexual predatory chats. *Forensic Science International: Digital Investigation*, 36, 301109.

- O'Day, D. R., & Calix, R. A. (2013). Text message corpus: Applying natural language processing to mobile device forensics. *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)* (pp. 1-6). California: IEEE.
- Okuno, S., Asai, H., & Yamana, H. (2015). A challenge of authorship identification for ten-thousand-scale microblog users. *2014 IEEE International Conference on Big Data (Big Data)* (pp. 52-54). Washington, DC: IEEE.
- Omar, A. H. (2014). Processing Malaysian Indigenous Languages: A Focus on Phonology and Grammar. *Open Journal of Modern Linguistics*, 728-738.
- Orebaugh, A., & Allnutt, J. (2009). Classification of instant messaging communications for forensics analysis. *The International Journal of Forensic Computer Science*, 22-28.
- Park, M. S.-A., Golden, K. J., Vizcaino-Vickers, S., Jidong, D., & Raj, S. (2021). Sociocultural Values, Attitudes and Risk Factors Associated With Adolescent Cyberbullying in East Asia: A Systematic Review. *Journal of Psychosocial Research on Cyberspace*. doi:<https://doi.org/10.5817/CP2021-1-5>
- Pawar, R., Agrawal, Y., Joshi, A., Gorrepati, R., & Raje, R. R. (2018). Cyberbullying detection system with multiple server configurations. *In 2018 IEEE International Conference on Electro/Information Technology (EIT)* (pp. 90-95). IEEE: Rochester.
- Perera, A., & Fernando, P. (2021). Accurate cyberbullying detection and prevention on social media. *Procedia Computer Science*, 181, 605-611.
- Posadas-Durán, J.-P., Gómez-Adorno, H., Sidorov, G., Batyrshin, I., Pinto, D., & Chanona-Hernández, L. (2017). Application of the distributed document representation in the authorship attribution task for small corpora. *Soft Computing* 21, 627–639.

- Potthast, M., Rosso, P., Stamatatos, E., & Stein, B. (2019). A Decade of Shared Tasks in Digital Text Forensics at PAN. *Lecture Notes in Computer Science*, 291-300.
- Punchihewa, M., Rajapaksha, C., & Asanka, D. (2021). A Language Modelling Approach to Authorship Identification for Online Examinations in Sinhala. *2021 International Conference on Advanced Research in Computing (ICARC-2021)* (pp. 66-69). Belihuloya, Sri Lanka: IEEE.
- Rocha, A., Scheirer, W. J., Forstall, C. W., Cavalcante, T., Theophilo, A., Shen, B., Stamatatos, E. (2017). Authorship Attribution for Social Media Forensics. *IEEE Transactions on Information Forensics and Security*, 5-33.
- Sachowski, J. (2019). *Implementing digital forensic readiness: From reactive to proactive process*. Massachusetts: Elsevier.
- Sari, Y., Stevenson, M., & Vlachos, A. (2018). Topic or Style? Exploring the Most Useful Features for Authorship Attribution. *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 343-353). New Mexico: ACL.
- Sarwar, R., & Hassan, S.-U. (2021). UrduAI: Writeprints for Urdu authorship identification. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 1-18.
- Savoy, J. (2020). Machine Learning Models. *Machine Learning Methods for Stylometry*, 109-151.
- Schwartz, R., Tsur, O., Rappoport, A., & Koppel, M. (2013). Authorship Attribution of Micro-Messages. *Proceedings of the 2013 on Empirical Methods in Natural Language Processing (EMNLP 2013)* (pp. 1880 - 1891). Seattle: ACL.

- Shrestha, P., Sierra, S., González, F. A., Rosso, P., Montes-y-Gómez, M., & Solorio, T. (2017). Convolutional Neural Networks for Authorship Attribution of Short Texts. *15th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 669-674). Spain: Association for Computational Linguistics.
- Soler-Company, J., & Wanner, L. (2017). On the Relevance of Syntactic and Discourse Features for Author Profiling and Identification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 681–687). Valencia, Spain: Association for Computational Linguistics.
- Stamatatos, E. (2008). Author Identification: Using Text Sampling to Handle the Class Imbalance Problem. *Information Processing and Management*, 790-799.
- Stamatatos, E. (2009). A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 538-556.
- Stamatatos, E. (2017). Authorship attribution using text distortion. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers* (pp. 1138-1149). Spain: ACL.
- Theophilo, A., Giot, R., & Rocha, A. (2021). Authorship Attribution of Social Media Messages. *IEEE Transactions on Computational Social Systems*, 1-14.
- Theóphilo, A., Pereira, L. A., & Rocha, A. (2019). needle in a haystack? harnessing onomatopoeia and user-specific stylometrics for authorship attribution of micro-messages. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2692-2696). Brighton: IEEE.
- Theóphilo, A., Pereira, L. A., & Rocha, a. A. (2019). A Needle in a Haystack? Harnessing Onomatopoeia and User-specific Stylometrics for Authorship Attribution of Micro-

- messages. *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2692-2696). Brighton: IEEE.
- Bhargava, M., Mehndiratta, P., & Asawa, K. (2013). Stylometric analysis for authorship attribution on twitter. *In International Conference on Big Data Analytics* (pp. 37-47). Cham: Springer.
- Vajjala, S., & Banerjee, S. (2017). A study of N-gram and Embedding Representations for Native Language Identification. *In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 240-248). Copenhagen: Association for Computational Linguistics.
- Yülüce, İ., & Dalkılıç, F. (2022). Author Identification with Machine Learning Algorithms. *International Journal of Multidisciplinary Studies and Innovative Technologies*, 45-50.
- Zainudin, N. M., Zainal, K. H., Hasbullah, N. A., Wahab, N. A., & Ramli, S. (2016). A review on cyberbullying in Malaysia from digital forensic perspective. *2016 International Conference on Information and Communication Technology (ICICTM)* (pp. 246-250). Kuala Lumpur: IEEE.
- Zhang, R., Hu, Z., Guo, H., & Mao, a. Y. (2018). Syntax encoding with application in authorship attribution. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 2742-2753). Brussels: Association for Computational Linguistics.
- Zheng, R., Yi Qin, Z. H., & Chen, H. (2003). Authorship analysis in cybercrime investigation. *In International Conference on Intelligence and Security Informatics* (pp. 59-73). Berlin: Springer.

APPENDICES

Appendix A

i) English Feedlists

assmuncher	assnigger	asspirate	assshit	assshole	asssucker
asswad	asswipe	axwound	bampot	bastard	beaner
bitch	bitchass	bitches	bitchtits	bitchy	blow job
blowjob	bollocks	bollox	boner	brotherfucker	bullshit
bumblefuck	butt plug	buttfucka	buttfucker	camel toe	carpetmuncher
chesticle	chinc	chink	choad	chode	clit
clitface	clitfuck	clusterfuck	cock	cockass	cockbite
cockburger	cockface	cockfucker	cockhead	cockjockey	cockknoker
cockmaster	cockmongler	cockmongruel	cockmonkey	cockmuncher	cocknose
cocknugget	cockshit	cocksmith	cocksmoke	cocksmoker	cocksniffer
cocksucker	cockwaffle	coochie	coochy	coon	cooter
cracker	cum	cumbubble	cumdumpster	cumguzzler	cumjockey
cumslut	cumtart	cunnie	cunnilingus	cunt	cuntass
cuntface	cunthole	cuntlicker	cuntrag	cuntslut	dago
damn	deggo	dick	dickbag	dickbeaters	dickface
dickfuck	dickfucker	dickhead	dickhole	dickjuice	dickmilk
dickmonger	dicks	dickslap	dicksucker	dicksucking	dicktickler
dickwad	dickweasel	dickweed	dickwod	dike	dildo
dipshit	doochbag	dookie	douche	douchebag	douchewaffle
dumass	dumb ass	dumbass	dumbfuck	dumbshit	dumshit
dyke	fag	fagbag	fagfucker	faggit	faggot
faggotcock	fagtard	fatass	fellatio	feltch	flamer
fuck	fuckass	fuckbag	fuckboy	fuckbrain	fuckbutt
fuckbutter	fucked	fucker	fuckersucker	fuckface	fuckhead
fuckhole	fuckin	fucking	fucknut	fucknutt	fuckoff
fucks	fuckstick	fucktard	fucktart	fuckup	fuckwad
fuckwit	fuckwitt	fudgepacker	gay	gayass	gaybob
gaydo	gayfuck	gayfuckist	gaylord	gaytard	gaywad
goddamn	goddamnit	gooch	gook	gringo	guido
handjob	hard on	heeb	hell	ho	hoe
homo	homodumbshit	honkey	humping	jackass	jagoff
jap	jerk off	jerkass	jigaboo	jizz	jungle bunny
junglebunny	kike	kooch	kootch	kraut	kunt
kyke	lameass	lardass	lesbian	lesbo	lezzie
mcfagget	mick	minge	mothafucka	mothafuckin\'	motherfucker
motherfucking	muff	muffdiver	munging	negro	nigaboo

nigga	nigger	niggers	niglet	nut sack	nutsack
paki	panooch	pecker	peckerhead	penis	penisbanger
penisfucker	penispuffer	piss	pissed	pissed off	pissflaps
polesmoker	pollock	poon	poonani	poonany	poontang
porch monkey	porchmonkey	prick	punanny	punta	pussies
pussy	pussylicking	puto	queef	queer	queerbait
queerhole	renob	rimjob	ruski	sand nigger	sandnigger
schlong	scrote	shit	shitass	shitbag	shitbagger
shitbrains	shitbreath	shitcanned	shitcunt	shitedick	shitface
shitfaced	shithead	shithole	shithouse	shitspitter	shitstain
shitter	shittiest	shitting	shitty	shiz	shiznit
skank	skeet	skullfuck	slut	slutbag	smeg
snatch	spic	spick	splooge	spook	suckass
tard	testicle	thundercunt	tit	titfuck	tits
tittyfuck	twat	twatlips	twats	twatwaffle	unclefucker
va-j-j	vag	vagina	vajayjay	vjayjay	wank
wankjob	wetback	whore	whorebag	whoreface	wop

ii) *KadazanDusun* Feedlist

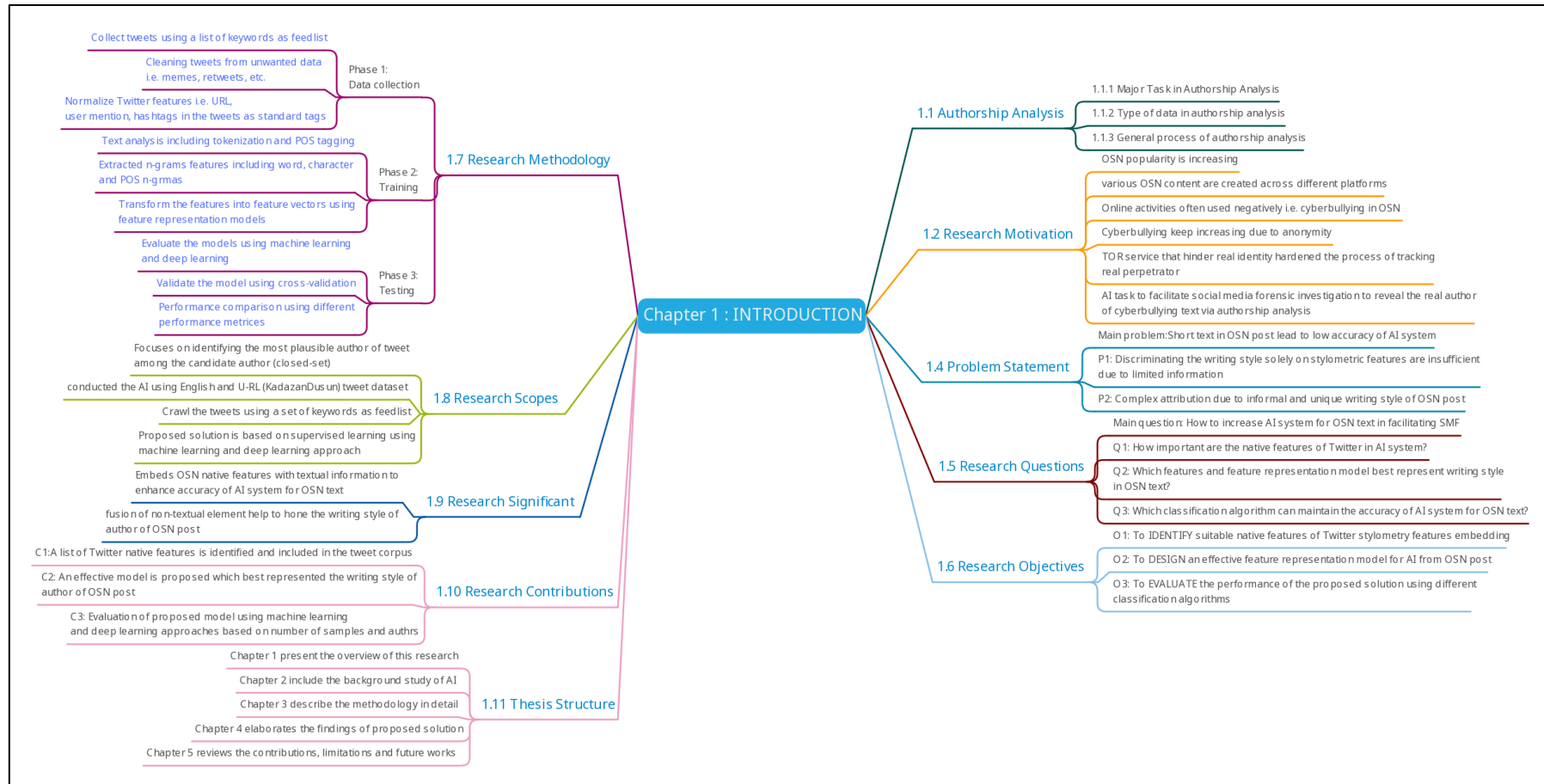
toburus
mulau
pukindet
basug
bantaton
basug
noruan
panansawa
apatai ko
baa
lohuhubang
kopio
tanak do
mulau
budu
tanak do
rogun
pukitai

Appendix B

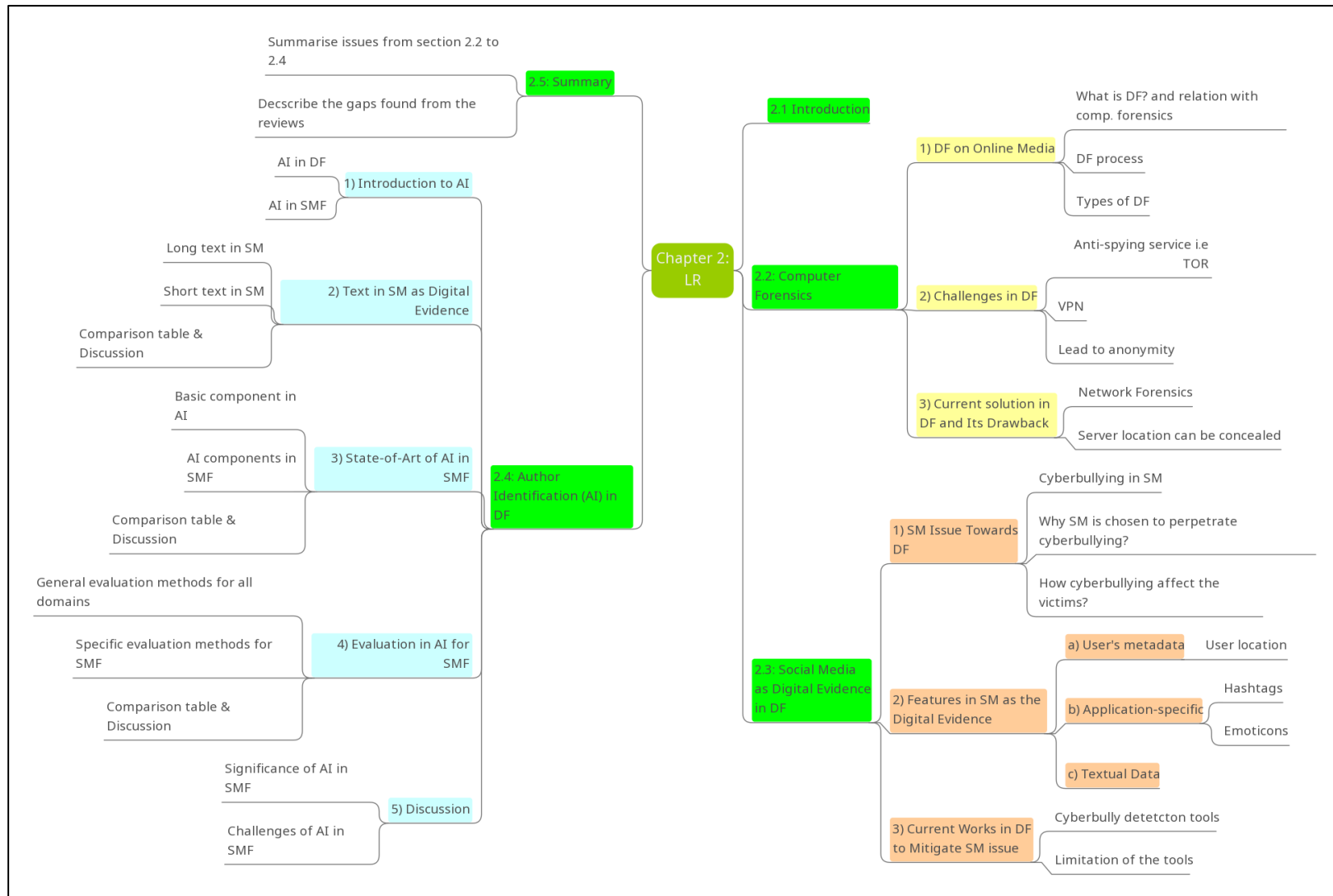
Description	Value
Total users	50
Total num. of tweets collected	69,624
Average num. of tweets per author	1,392 tweets per author
Total number of vulgar word occurrences	29,701
The average number of vulgar word occurrences per tweet	Two words per tweet
Max num. of vulgar word occurrences from an author	3187 words
Min. num. of vulgar word occurrences from an author	9 words

Appendix C

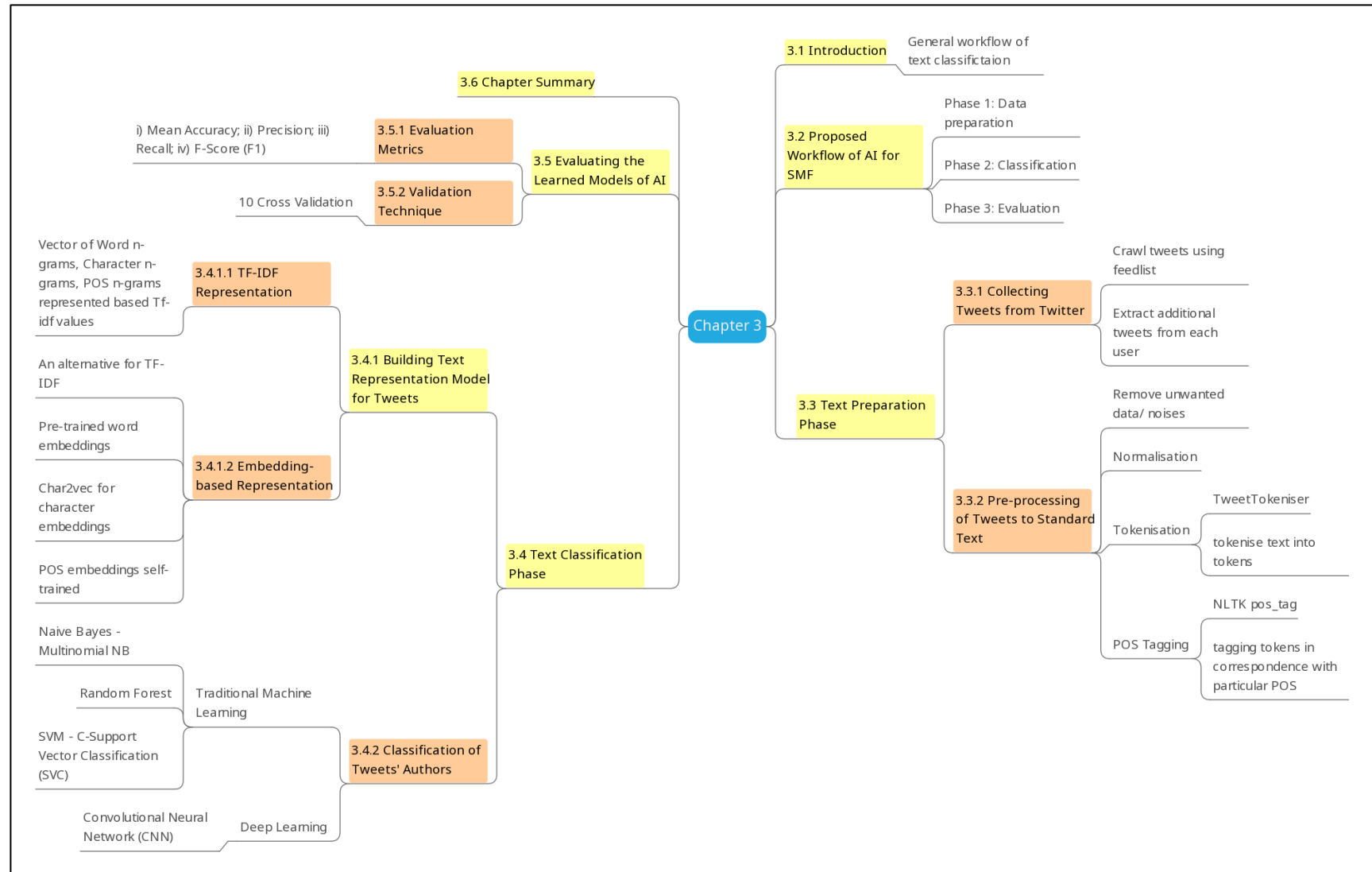
i) Chapter 1



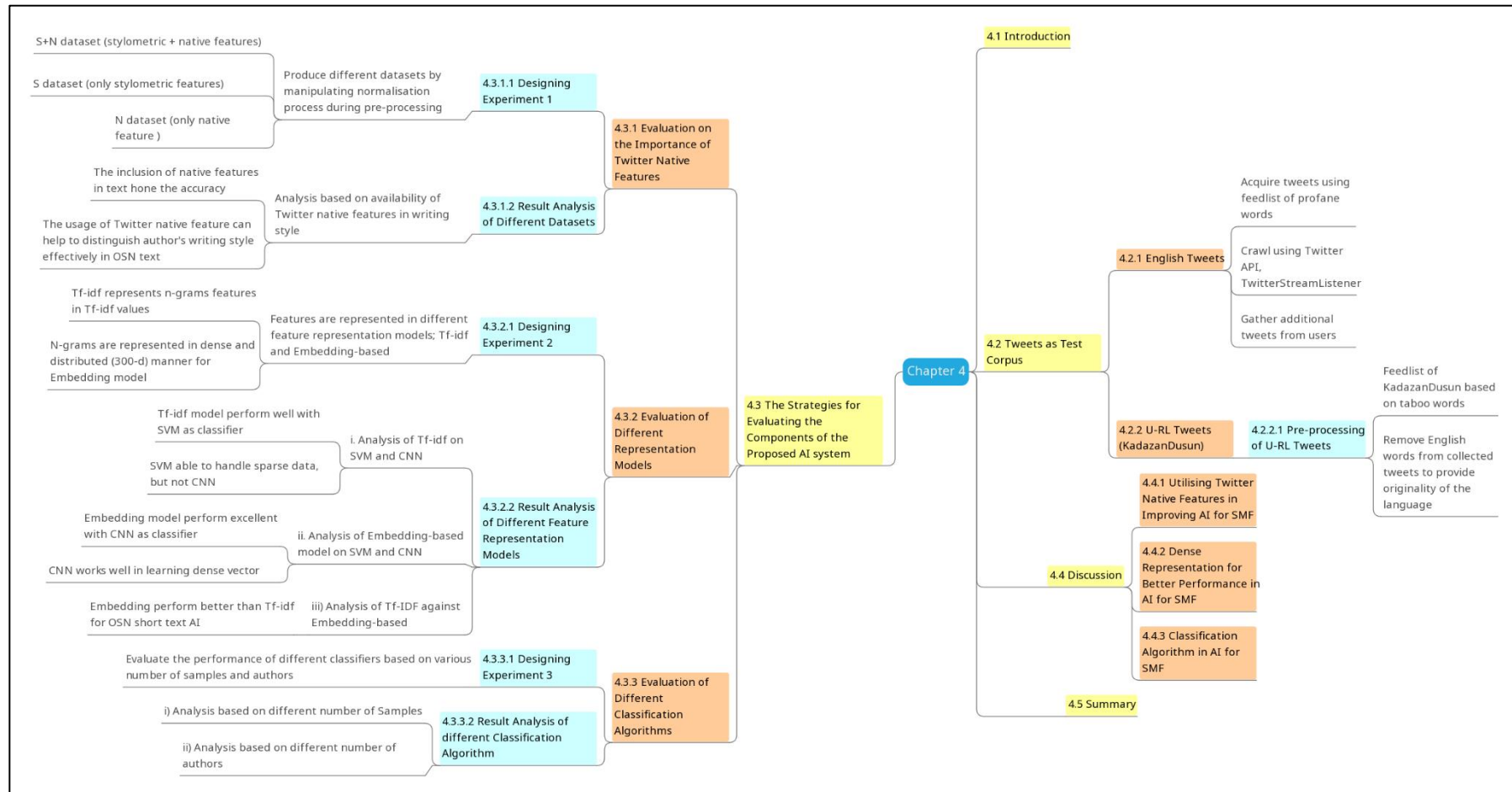
ii) Chapter 2



iii) Chapter 3



iv) Chapter 4



v) Chapter 5

