University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

5-2014

Text stylometry for chat bot identification and intelligence estimation.

Nawaf Ali
University of Louisville

Follow this and additional works at: <https://ir.library.louisville.edu/etd>

Part of the [Computer Engineering Commons](#)

Recommended Citation

Ali, Nawaf, "Text stylometry for chat bot identification and intelligence estimation." (2014). *Electronic Theses and Dissertations*. Paper 31.
<https://doi.org/10.18297/etd/31>

This Doctoral Dissertation is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

TEXT STYLOMETRY FOR CHAT BOT IDENTIFICATION AND INTELLIGENCE ESTIMATION

By

Nawaf Ali

B.Sc. IT, Al-Balqa Applied University, Jordan, 2001

M.S., Al-Balqa Applied University, Jordan, 2005

A Dissertation

Submitted to the Faculty of the J. B. Speed School of Engineering of the University
of Louisville

In Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

Department of Computer Engineering and Computer Science
University of Louisville
Louisville, Kentucky

May 2014

Copyright© 2014 by Nawaf Ali

All rights reserved

TEXT STYLOMETRY FOR CHAT BOT IDENTIFICATION AND INTELLIGENCE ESTIMATION

By

Nawaf Ali

B.Sc. IT, Al-Balqa Applied University, Jordan, 2001

M.S., Al-Balqa Applied University, Jordan, 2005

A Dissertation Approved on

April 18th, 2014

By the following Dissertation Committee:

Dr. Roman V. Yampolskiy, CECS Department, Dissertation Director

Dr. Adel S. Elmaghraby, CECS Department Chair

Dr. Ibrahim N. Imam, CECS Department

Dr. Dar-jen Chang, CECS Department

Dr. Charles Timothy Hardin, Speed Industrial Engineering

DEDICATION

This dissertation is dedicated to my parents and my wife who had been loving and patient throughout the way; I could never have done this without your blessing and support.

Eng. Mohammad Abu Taha, who gave me the initiative to start, and supported me along the way.

My Brothers, Sisters and all my Friends.

ACKNOWLEDGMENT

It is my greatest pleasure to thank my mentor Dr. Roman V. Yampolskiy. I owe this achievement to his insight and inspiration. He motivated us through his ideas and solutions and was always there for my colleagues and myself. I appreciate what he has done for me and wish that he accomplishes all that he dreams of.

Dr. Elmaghraby, you were always like a big brother to us and would never let us down in times of need. Being the first person I knew here, you always supported me. I sincerely appreciate all your efforts to help me and the department.

Dr. Imam, I remember the times when you taught me. I did learn a lot from your classes and enjoyed the experience.

Dr. Chang, thanks for your advice and support during my course work and research work. I am so grateful for all that you have done. Thank you.

Dr. Hardin, your class is a real-life experience. I enjoyed your style of teaching and would definitely benefit from that in my academic career. I appreciate your constant help, support and guidance throughout my academic experience.

My wife Muna, you were the energy fueling my body and soul. It would have been impossible to achieve this without you by my side. You are the woman that I have dreamt of. You are very supportive, passionate and loving. I failed to think of anything that could be enough to pay you back. I will give my best to make you the happiest woman in the universe. May God help me accomplish this with our beautiful and wonderful kids.

Shahrazad, my precious daughter and first child, you are an extremely passionate and a loving kid. I pray that you will accomplish all your dreams.

Sief, my first baby boy, I hope that you will grow up strong and smart as you always have been. I pray that you too will accomplish all your dreams.

Malak, my little angel girl, I love you so much. I hope that you will be like your mother.

Finally, my little one, the champ, Muhammad Ali, you bring the joy and happiness to all of us. We love you so much.

My parents, I wish that I could get the chance to pay back a part of your sacrifices for me. I wish you be healthy and have a long life for me to make it up to you.

ABSTRACT

**TEXT STYLOMETRY FOR CHAT BOT
IDENTIFICATION AND INTELLIGENCE
ESTIMATION**

Nawaf Ali

April, 18th, 2014

Authorship identification is a technique used to identify the author of an unclaimed document, by attempting to find traits that will match those of the original author. Authorship identification has a great potential for applications in forensics. It can also be used in identifying chat bots, a form of intelligent software created to mimic the human conversations, by their unique style. The online criminal community is utilizing chat bots as a new way to steal private information and commit fraud and identity theft. The need for identifying chat bots by their style is becoming essential to overcome the danger of online criminal activities.

Researchers realized the need to advance the understanding of chat bots and design programs to prevent criminal activities, whether it was an identity theft or even a terrorist threat. The more research work to advance chat bots' ability to perceive humans, the more duties needed to be followed to confront those threats by the research community. This research went further by trying to study whether chat bots have behavioral drift.

Studying text for Stylometry has been the goal for many researchers who have experimented many features and combinations of features in their experiments. A novel feature has been proposed that represented Term Frequency Inverse Document

Frequency (TFIDF) and implemented that on a Byte level N-Gram. Term Frequency-Inverse Token Frequency (TF-ITF) used these terms and created the feature. The initial experiments utilizing collected data demonstrated the feasibility of this approach. Additional versions of the feature were created and tested for authorship identification. Results demonstrated that the feature was successfully used to identify authors of text, and additional experiments showed that the feature is language independent. The feature successfully identified authors of a German text. Furthermore, the feature was used in text similarities on a book level and a paragraph level. Finally, a selective combination of features was used to classify text that ranges from kindergarten level to scientific researches and novels. The feature combination measured the Quality of Writing (QoW) and the complexity of text, which were the first step to correlate that with the author's IQ as a future goal.

Keywords – Authorship identification, Chat bots, Stylometry, Text mining, Behavioral drift, Biometrics, N-Gram, TFIDF, TF-ITF, BLN-Gram-TF-ITF, Text similarity, Quality of Writing.

TABLE OF CONTENTS

DEDICATION	iii
ACKNOWLEDGMENT	iv
ABSTRACT	vi
TABLE OF CONTENTS	viii
LIST OF FIGURES	x
LIST OF TABLES.....	xi
CHAPTER I.....	1
INTRODUCTION.....	1
1.1 Biometrics	1
1.2 Authorship Identification History	2
1.3 Text Features Used in Stylometry.....	4
1.3.1 Lexical Features.....	4
1.3.2 Character Features.....	6
1.3.3 Syntactic Features	7
1.3.4 Semantic Features	8
1.4 Related Work.....	9
1.4.1 Identifying Chat bots	12
1.4.2 Chat bots Implementation Algorithms	12
1.4.3 Using Stylometry to tell Chat bots from Humans	13
1.4.4 Features that include inter and intra-paragraph stylistic issues.....	14
1.4.5 Metrics that are best for similarity rather than for exclusion of authorship.....	15
1.4.6 Metrics that are best for exclusion rather than for similarity of authorship.....	16
1.4.7 Text Classification	17
1.5 Authorship versus Text Similarity	19
1.6 Quality of Writing (QoW) as an IQ Measure	20
1.7 Conclusion	20
CHAPTER II	22
PRELIMINARY WORK	22
2.1 Chat Bots	22
2.2 Turing Test.....	22
2.3 Motivation	23
2.4 Data Collection.....	24

2.5 Applications Used for Testing	25
a. Stylometry Project	25
b. Java Graphical Authorship Attribution Project (JGAAP)	26
.....	27
2.6 Chat bots used	29
2.7 Behavioral Drift.....	30
2.8 Content Analyses and the Other IQ.....	31
2.9 Term Frequency Inverse Document Frequency (TFIDF) measure	31
2.10 Byte-Level N-Gram Term Frequency Inverse Token Frequency.....	33
2.11 Conclusion	35
CHAPTER III	36
EXPERIMENTS AND RESULT ANALYSIS.....	36
3.1 Detecting Authorship	36
3.2 Detecting Behavioral Drift.....	40
3.2.1 Data Collection and Preparation	40
3.2.2 Behavioral Drift Experiments.....	42
3.3 Byte-Level N-Gram Term Frequency Inverse Token Frequency	47
3.3.1 Data used.....	47
3.3.2 Matlab Classification	48
3.3.3 Classification Results.....	48
3.4 Authorship Identification using BLN-Gram-TF-ITF.....	49
3.4.1 Experimenting on Chat bot Corpus	49
3.4.2 Experimenting on Human Corpus	50
3.4.3 BLN-Gram-TF-ITF for Paragraph similarity	55
3.4.4 Experiments with Combining BOW with BLN-Gram-TF-ITF	56
3.4.5 BLN-Gram-TF-ITF as Language Independent Feature.....	58
3.5 Bitcoins Mysterious Founder	59
3.6 Quality of Writing (QoW) Features.....	60
3.6.1 Features used for Quality of Writing (QoW)	61
3.6.2 Scientific Writing Samples versus School Students Writing.....	61
3.6.3 Classifying Three Classes.....	61
3.6.4 Classifying Six Classes.....	62
CHAPTER IV	64
CONCLUSION AND FUTURE WORK	64
5.1 Authorship Identification and the BLN-Gram-TF-ITF.....	64
5.2 Quality of Writing versus IQ.....	65
REFERENCES.....	66
Appendix A.....	73
Appendix B.....	74
Appendix C	76
Appendix D	77
CURRICULUM VITAE	79

LIST OF FIGURES

FIGURE 1: TOKEN-BASED N-GRAM EXAMPLE FOR N=3	6
FIGURE 2: CHARACTER N-GRAM FOR N=3	7
FIGURE 3: PERCENTAGE (Y-AXIS) OF SNIPPETS FOR WHICH THE ACTUAL AUTHOR IS ASSIGNED A RANK K OR BETTER (X-AXIS).....	11
FIGURE 4: UNMASKING "AN IDEAL HUSBAND" AGAINST TEN AUTHORS	19
FIGURE 5: COSINE SIMILARITY USING VECTOR REPRESENTATION	20
FIGURE 6: THE CHAT BOT RELOADED FLOW CHART.....	24
FIGURE 7: SAMPLE CONVERSATION BETWEEN CHAT BOTS	25
FIGURE 8: DOCUMENTS LOADING STAGE IN JGAAP	27
FIGURE 9: CANONICIZATION STAGE IN JGAAP	27
FIGURE 10: EVENT SET OR FEATURE SELECTION STAGE IN JGAAP	28
FIGURE 11: ANALYZE OR CLASSIFICATION STAGE IN JGAAP	28
FIGURE 12: REPORT STAGE IN JGAAP	29
FIGURE 13: CHAT BOTS USED IN THE RESEARCH TAKEN FROM CHAT BOTS' WEBSITE	31
FIGURE 14: FLOW CHART OF THE BYTE-N-GRAM-TF-ITF	35
FIGURE 15: FLOW CHART OF THE PROCESS FOLLOWED DURING THE EXPERIMENTS	37
FIGURE 16: AVERAGE ACCURACY FOR EACH CLASSIFIER	38
FIGURE 17: AVERAGE ACCURACY FOR EACH FEATURE	39
FIGURE 18: RAW CHAT DATA BEFORE CLEANING FROM THE LOEBNER CONTEST	41
FIGURE 19: CHAT BOT SAMPLE CHAT LOG AFTER SEPARATION AND CLEANING	41
FIGURE 20: THE MODEL USED FOR TRAINING IN RM	42
FIGURE 21: PREPROCESSING STAGE	43
FIGURE 22: WORDLIST OR BAG OF WORD (BOW) SAVED.....	43
FIGURE 23: TESTING MODEL USED IN RM	43
FIGURE 24: PREDICTION LEVEL OVER YEARS UNDER STUDY	46
FIGURE 25: CONFUSION MATRICES FOR ONE RUN OF THE EXPERIMENTS	49
FIGURE 26: BLN-GRAM-TF-ITF VS. JGAAP AVERAGE ACCURACY AND MAXIMUM ACCURACY FOR CHAT BOT'S CORPUS	50
FIGURE 27: FILE SIZE IN WORD COUNTS VS. AVERAGE ACCURACY ACHIEVED FOR N=3	52
FIGURE 28: FILE SIZE IN WORD COUNTS VS. AVERAGE ACCURACY ACHIEVED FOR N=4	52
FIGURE 29: FILE SIZE IN WORD COUNTS VS. AVERAGE ACCURACY ACHIEVED FOR N=5	53
FIGURE 30: FILE SIZE IN WORD COUNTS VS. AVERAGE ACCURACY ACHIEVED FOR N=3, 4, AND 5.....	54
FIGURE 31: BOX AND WHISKER PLOT FOR THE AVERAGE ACCURACY FOR N=3, 4, AND 5....	54
FIGURE 32: THE BYTE LEVEL N-GRAM TERM FREQUENCY INVERSE TOKEN FREQUENCY COMBINED WITH BOW FLOW CHART	57
FIGURE 33: AVERAGE ACCURACY VS. FILE SIZE BETWEEN BLN-GRAM-TF-ITF AND COMBINED WITH BOW	58

LIST OF TABLES

TABLE 1: LEXICAL FEATURES' DESCRIPTION	5
TABLE 2: LEXICAL FEATURES WITH THEIR CORRESPONDING REQUIRED TOOLS.....	6
TABLE 3: CHARACTER FEATURES WITH THEIR CORRESPONDING REQUIRED TOOLS.....	7
TABLE 4: SYNTACTIC FEATURES WITH THEIR CORRESPONDING REQUIRED TOOLS	8
TABLE 5: SIMPLIFIED PART -OF-SPEECH TAG SET [39].....	8
TABLE 6: THE SEMANTIC FEATURES WITH THEIR CORRESPONDING REQUIRED TOOLS.....	9
TABLE 7: AVERAGE ACCURACY FOR EACH FEATURE OVER ALL CLASSIFIERS	39
TABLE 8: ALICE'S CONFIDENCE LEVELS	44
TABLE 9: ALICE CONFIDENCE LEVELS WITH THREE BOTS TRAINING SET.....	44
TABLE 10: JABBERWACKY'S PREDICTION VALUES WHEN TRAINING WITH ALL BOTS.....	45
TABLE 11: JABBERWOCK'S CONFIDENCE VALUES TRAINING WITH ALL BOTS.....	45
TABLE 12: T-TEST RESULTS FOR STATISTICAL SIGNIFICANCE BETWEEN JGAAP AVERAGE ACCURACY AND BLN-GRAM-TF-ITF AVERAGE ACCURACY	50
TABLE 13: SIMILARITIES BETWEEN PARAGRAPHS FROM SIX DIFFERENT BOOKS	55
TABLE 14: PAIRED T-TEST FOR STATISTICAL SIGNIFICANT BETWEEN THE TWO AVERAGE ACCURACIES OF BLN-GRAM-TR-ITF AND THE COMBINED FEATURE WITH BOW. 95% CI FOR MEAN DIFFERENCE :(-0.1314, -0.0362)	57
TABLE 15: SIMILARITIES BETWEEN CANDIDATE AUTHORS	59
TABLE 16: SIX CLASSES PREDICTION USING (QOW) FEATURES.....	62

CHAPTER I

INTRODUCTION

1.1 Biometrics

Biometrics is the science of analyzing human characteristics by using automated methods in a way that gives every person an identity, making him or her unique from others. Biometric identification is a way to discover or verify the identity of who we claim to be, by using physiological or behavioral traits [1]. Are you who you are claiming to be, or in actuality are you someone else? To better serve as a successful identifier, a biometric should have the following properties:

- a) Universality: A biometric characteristic should apply to everyone.
- b) Uniqueness: A biometric characteristic should be unique for each individual.
- c) Permanence: The characteristics should not change over time in a way that will obscure the identity of a person.
- d) Collectability: The ability to collect such characteristics [2].

Establishing the identity of a person is becoming critical in our vastly interconnected societies. The need for a reliable user authentication techniques has increased in the wake of heightened concerns about security and rapid advancement in networking [3].

Behavioral biometrics provides a number of advantages over traditional (Physiological) biometric technologies. They can be collected non-obtrusively or even without the knowledge of the user [1]. Yampolskiy et al. stated that different types of Biometrics include, but not limited to:

- A. Physiological: Like Fingerprints, Hand Scans (include knuckle, palm, and Vascular), Retina Scan, Iris Scan, Facial Scan, DNA, Odor, Earlobe, Sweat pore, Lips, Thermo grams, Vein Patterns, Ear recognition, and Skin Reflection.

B. Behavioral: Like Voice, Speech, Singing, Keystroke dynamics, Signature, Handwriting, Gait, Stride, Audit logs, Biometric sketch, Blinking, Call-stack, Calling behavior, Car driving style, Command line lexicon, Credit card use, Dynamic facial features, E-mail behavior, Game strategy, GUI interaction, Handgrip, Haptic, Lip Movement, Mouse dynamics, Network traffic, Painting style, Programming style, Registry access, Soft behavioral biometrics, Storage activity, System calls, Tapping, and Text authorship [4].

Human usage of language, writing, set of vocabulary, unusual usage of words, and particular syntactic and stylistic traits tend to be stable. The big challenge for authorship identification is locating and learning from such traits [5].

1.2 Authorship Identification History

Authorship identification is one of the oldest Information Retrieval (IR) problems. This problem has been known since the first document was created and remains an active area of research, with many potential applications in forensics.

By definition, authorship identification is the science of identifying the author of an unclaimed document. Writers always leave an “authorial fingerprint” [6] in their writings, even if they intended not to do so. This “fingerprint” is called the style [2].

In the 19th century, an unprecedented research was performed by Mendenhall in which he attempted to quantify styles by investigating the Shakespeare’s plays [7]. Later on in 1932 Zipf [8], and Yule 1938 [9], and Yule 1944 [10] started the first statistical analysis on text with the purpose of trying to find any distinguishing features associated with each author. Zipf’s law states that for each word with frequency f and rank r , sorted from highest frequency word to lowest, then $(f * r)$ will yield a constant value over all terms.

In 1964, Mosteller and Wallace were the first to use computational and statistical fundamentals for the purpose of authorship analysis [11]. Their work on the 146 political essays, known as the Federalist Papers and written by Alexander Hamilton, John Jay, and James Madison, revealed that twelve essays could be

attributed to Hamilton and Madison through the implementation of Bayesian statistical analysis of the frequencies of the common words, or what will later be known as function words [12].

Mosteller and Wallace's creative work was the spark that started and spread authorship research using non-traditional techniques, by defining feature for quantifying writing style, which is now known as Stylometry [13].

Due to the increased usage of the Internet and the large availability of online electronic texts and media (blogs, emails, forums, etc.) seen since the 1990's, the need for authorship identification studies to handle these files is growing daily.

The immerging scientific areas of Information Retrieval (IR) and Natural Language Processing (NLP) have boosted the authorship attribution research by a great deal. Powerful machine learning algorithms have enabled researchers to deal with large and sparse data [12, 14].

From the machine learning perspective, authorship identification problem is defined as multi-class single-label text categorizing problem [15]. Having the disputed text, we need to assign this text to one of the different authors claiming this text.

In general, authorship identification tasks can be categorized under the following:

- Authorship verification: Given a certain text, we need to verify if it was written by a certain author or not [2].
- Plagiarism detection: Looking for copied materials from one source to another without being referenced by finding similarities between the two texts.
- Author profiling or characterization: Extracting author information from his text.
- Authorship recognition: Determining the right author of a text from several potential authors.
- Detection of stylistic inconsistencies: When more than one author participate in writing the text [12].

1.3 Text Features Used in Stylometry

Defining and extracting features from text was first proposed by Mosteller and Wallace in 1964 in their work on the Federalist papers [16]. Sentence length, word length, word frequencies, character frequencies, and vocabulary richness functions were proposed to solve the authorship identification problem [6]. Around 1,000 different features have been studied, but the methodologies used were computer assisted and not computer based, meaning that the process was not fully automated [12].

1.3.1 Lexical Features

Text can be broken into tokens, with each token representing a word, number, or punctuation mark. Using this approach to calculate simple measures, such as sentence length count and word length count, is the function of lexical features [7]. Due to the reliance of this method on the tokens, lexical features are language independent. You simply need a tokenize function to be able to work with different languages, with the exception of the certain natural languages like Chinese.

The Vocabulary Richness measure is a measure for the ratio of unique tokens in the text. Type-token ratio is an example of such measure.

$$\text{Type Token Ratio} = \frac{V}{N} \quad (1)$$

Where V is the number of unique tokens, and N is the total number of tokens in text.

Table 1 lists some of the examples used for lexical features in Java Graphical Authorship Attribution Project (JGAAP) [17]. Vectors representing word frequencies, also known as Bag Of Words (BOW) approach, is one of the most straightforward approaches used to represent text [18, 19].

Table 2 shows lexical features and the corresponding tools needed for their measurement. Function words, which are the most common words used in writing (articles, prepositions, pronouns, etc.) was found to be one of the best features used to discriminate between authors. Stop words is another name for these kinds of words,

while these words should be eliminated when testing for text similarities topic based categorization [12].

Topic based classification technique will calculate frequencies of tokens in documents, excluding function words. Due to the unconscious use of function words by authors, they are very useful stylistic features [19, 20].

Selection of function word(s) that will be used as a feature is usually based on arbitrary criteria. This process mainly will require a linguistics expert. Various researchers claimed different sizes of their function words set, Abbasi and Chen have 150 words in their set [21], while Argamon et al. have 303 words in their set [22]; Zhao et al. has set of 365 function words [23]; 480 function words were proposed by Koppel et al. [24]; another set of 675 words was reported by Argamon et al. [25]

Table 1: Lexical Features' Description

Feature Used	Feature usage description
2-3 Letters	Words with 2 or 3 letters length.
2-4 Letters	Words with 2, 3, or 4 letters length.
3-4 Letters	Words with 3 or 4 letters length.
Character Bigrams	Character pairs in sequence.
Characters	Unicode Characters frequencies.
Character Tetra Grams	Groups of four successive letters.
Character Trigrams	Groups of three successive letters.
Dis Legomena	Words appearing only twice in the document.
Hapax Legomena	Words appearing only once in the document.
Hapax-Dis Legomena	Words appearing once or twice in the document.
MW function Words	Function words from Mosteller-Wallace.
Words	Words frequencies (white space as separator).
Vowels 2-3 letters words	Words starting with a vowel with length of 2 or 3 letters.
Vowels 2-4 letters words	Words starting with a vowel with length of 2, 3, or 4 letters.
Vowels 3-4 letters word	Words starting with a vowel with length of 3 or 4 letters.
Vowel initial words	Words starting with a Vowel (A, E, I, O, U).
Word Bigrams	Word pairs in sequence.
Word Length	The length of words in each document.
Syllables per word	Number of vowel cluster per word.
Word Tetra Grams	Groups of four successive words.

Word N-Grams, also known as word collocations, are n consecutive words. It is one of the most useful features, appealing to many researchers [26-28]; it has proven to give more successful results when compared to other features [29-33]. Figure 1 shows an example N-Gram for $N=3$ which is commonly called a trigram.

1.3.2 Character Features

Text can also be studied as a sequence of characters and not as sequence of tokens. In this feature, letters frequencies instead of the word frequencies are used. Many other statistics can be performed on capital letters count, number counts, and punctuations [34, 35]. This feature is language independent, so it can be applied upon many languages, taking into consideration that character frequencies differ from language to language.

Table 2: Lexical Features with their corresponding required tools

Features		Required Tools
Lexical	Token based (word length, sentence length, etc.)	Tokenizer, (Sentence splitter)
	Vocabulary richness	Tokenize
	Word frequencies	Tokenizer, (Stemmer)
	Word N-gram	Tokenizer
	Errors	Tokenizer, orthographic spell checker

“I played with my brother this morning”

I played with	played with my	with my brother	my brother this	brother this morning
---------------	----------------	-----------------	-----------------	----------------------

Figure 1: Token-based N-Gram Example for $N=3$

Table 3: Character Features with their corresponding required tools

Features		Required Tools
Character	Character types (Letters, digits, etc.)	Character Dictionary
	Character n -grams (Fixed-length)	-
	Character n -grams (Variable-length)	Feature selector
	Compression methods	Text compression tool

As N-Gram was applied to tokens, it can also be applied to characters. Following the same technique as shown in Figure 2, It was proven to be useful for quantifying writing style even for oriental languages where tokenization is a difficult obstacle [36].

Figure 2: Character N-Gram for N=3

A major factor in the success or failure of the N-Gram technique is the selection of N value. The corpus used and language of text will affects the selection on N [37].

1.3.3 Syntactic Features

Syntactic patterns are considered to possess more authorial traits than the lexical patterns [12].

Baayen et al. presented a simple approach called the Part-of-Speech (POS) Tagging that was very helpful measure for authorship attribution [38]. This is a method of assigning a tag to each word as a kind of functional description of that word.

Table 4 shows the syntactic features with the required tool to extract each feature.

Table 4: Syntactic Features with their corresponding required tools

	Features	Required Tools
Syntactic	Part-of-Speech	Tokenizer, Sentence splitter, POS tagger
	Chunks	Tokenizer, Sentence splitter, [POS tagger], Text chunker
	Sentence and phrase structure	Tokenizer, Sentence splitter, POS tagger, Text chunker, Partial parser
	Rewrite rules frequencies	Tokenizer, Sentence splitter, POS tagger, Text chunker, Full parser
	Errors	Tokenizer, Sentence splitter, Syntactic spell checker

Table 5: Simplified Part -of-Speech Tag set [39]

Tag	Meaning	Examples
ADJ	Adjective	new, good, high, special, big, local
ADV	Adverb	really, already, still, early, now
CNJ	Conjunction	and, or, but, if, while, although
DET	Determiner	the, a, some, most, every, no
EX	Existential	there, there's
FW	Foreign word	dolce, ersatz, esprit, quo, maitre
MOD	Modal verb	will, can, would, may, must, should
N	Noun	year, home, costs, time, education
NP	Proper noun	Alison, Africa, April, Washington
NUM	Number	twenty-four, fourth, 1991, 14:24
PRO	Pronoun	he, their, her, its, my, I, us
P	Preposition	on, of, at, with, by, into, under
TO	The word to	to
UH	Interjection	ah, bang, ha, whee, hmpf, oops
V	Verb	is, has, get, do, make, see, run
VD	Past tense	said, took, told, made, asked
VG	Present participle	making, going, playing, working
VN	Past participle	given, taken, begun, sung
WH	wh determiner	who, which, when, what, where, how

Table 5 shows a list of a Simplified POS Tag set. POS did show a good performance when dealing with unrestricted text, and many research works found that using POS tag frequencies or POS tag n-gram to represent style were very effective [24, 40-44].

1.3.4 Semantic Features

Current NLP technology does not show a great success when trying to perform semantic analysis for unrestricted text. This leads to less efforts targeted toward

extracting semantic features for the purpose of authorship attribution [12]. Published research by Leskovec, et al. claimed successful results by implementing an algorithm to connect documents using trees or graphs [45]. Table 6 shows the semantic features with their corresponding tool needed to extract that feature.

Gamon et al. made a claim of producing a tool to obtain semantic dependency graphs. No published reports have been made on the accuracy rate of their findings [43].

Argamon et al. performed perhaps the most impressive work by the introduction of semantic information using the theory of Systematic Functional Grammar (SFG) [25, 46].

Table 6: The semantic features with their corresponding required tools

Features		Required Tools
Semantic	Synonyms	Tokenizer, POS tagger, Thesaurus
	Semantic dependencies	Tokenizer, Sentence splitter, POS tagger, Text Chunker, Partial parser. Semantic parser
	Functional	Tokenizer, Sentence splitter, POS tagger, Specialized dictionaries

They relate definite words or phrases with semantic phrases using function features. Functional features did show an improvement in the classification outcomes if combined with stop words [12].

1.4 Related Work

Authorship attribution has been an active research field lately. The huge number of various social networks had dramatically increased the availability of online digital media. A need to know authors behind tweets, blogs and Facebook feeds had become an increasing interest for many researchers in vast range of applications, in stylometry, forensics, business campaigns, and many more [12].

Koppel et al. worked on identifying the owner of a blog with thousands of candidate authors by combining information retrieval methods with text categorization meta-learning schema. From authorship attribution perspective, one is given a closed set of authors and is required to decide which one of the authors is the candidate author for a disputed text. One can use text categorization to solve this problem. In real life, number of candidate authors is not a closed set; one would either have no closed set at all or a huge number of candidate authors, which makes the text categorization inappropriate [47].

Koppel et al. studied a set of 18,000 blogs. Each blog is a full set of 200 words posts by a given author. Gender distribution was equal for each age interval.

10,000 blogs were set for testing and the rest is set for training and validation. A minimum of 500 words from the chronological end of each of the 10,000 blogs is cut off and called “snippets”.

Three representations of texts were examined:

- Content tfidf: tfidf over content words.
- Content idf: binary idf over content words.
- Style tfidf: tfidf over stylistic features like function words.

The goal is to determine which author is the correct author for each of the 10,000 snippets; the cosine method was used for this purpose. Authors were ranked by similarities to the snippet. Figure 3 shows that over 20% of the snippets are most similar to the actual author. The previous result is never conclusive, and the results will be wrong most of the times, so meta learning scheme is used to exploit the 8,000 blogs which were set aside for learning purpose [47].

Figure 3: Percentage (y-axis) of snippets for which the actual author is assigned a rank k or better (x-axis)

Author with highest rank will be picked as the correct author of that snippet, and that pair will be called a “successful pair”. 18 different meta features were representing each pair, along with the absolute similarity of the snippet to the top-ranked. Linear SVM was used for classification [47].

The hypothesis for the 10,000 snippets using style-based features is 21.5% reliable to generate an overall 84.0% accuracy. For the Content tfidf, it is hypothesized for 25% of the snippets to generate an overall accuracy of 81.1%. Finally, a 34% of the snippets are hypothesized reliable with an overall accuracy of 79.7 for the content idf.

Combining the methods as follows, if one of the representation methods yields a top-ranked author, that author is labeled as the right author. If none of the three methods yield a success, the output will be a “Don’t know”. Finally, if two methods yield a different author with both hypothesized as reliable, we output “Don’t know”. Overall, with a 31.3% hypothesized reliable of the complete 10,000 snippets, the snippets are 88.2% correctly classified [47].

In conclusion, Koppel et al. decided that if dealing with a large number of candidate authors (thousands), one might accept the “Don’t know” as an answer. Despite that, they manage to achieve a reasonable reliable authorship attribution when the number of authors is in thousands [47].

1.4.1 Identifying Chat bots

“A Chat bot, Chatter bot, Chatter box or Chatter robot is a computer application designed to simulate a conversation with a human” [48]. Chat bots are mainly used in applications when human can be replaced with a bot to serve customers, such as online help, e-commerce, customer services, call centers, and internet gaming [49].

Chat bots are typically perceived as engaging software entities, which humans can talk to. Some Chat bots use sophisticated Natural Language Processing Systems (NLPS), but other types will just scan for keywords within the input and pull a reply with the most matching keywords [50]. Chat bots are still a largely developing technology; consequently, quality of simulated conversations varies from realistic to mostly nonsense [51].

Ali et al. experimented on data collected from chat bots talking to each other [52]. Java Graphical Authorship Attribution Project was used as an open source tool [17]. The reason behind that research was to investigate if chat bots have a consistent style. Ali et al. confirm that chat bots do have a unique style, and they manage to identify the correct chat bot using samples from chat bots’ texts [52]. Behavioral drift by definition is when author acquires new knowledge that will affect his style of writing. Ali et al. demonstrated this concept on chat bots by analyzing data from Loebner prize competition. The study did confirm that chat bot’s style drifts and changes as chat bots accumulate more knowledge [53].

1.4.2 Chat bots Implementation Algorithms

Many chat bots are designed to provide specific information and direct the user to a specific topic. Some chat bots recognize keywords; some recognize phrases and others handle whole sentences. There are dozens of different bots, developed for a variety of reasons. They range from hardwired programs with simply coded patterns to systems built upon embedded learning algorithms, which continuously expand their language knowledge-base using sophisticated Natural Language Processing Systems and employing learning algorithms. That gives the bot the means to remember information and use it in future responses. This enables chat bot to expand

its knowledge by learning new sentences from its users. The bot learns whole phrases, words and constantly expands its capabilities [54].

In general, chat bots are designed and built to follow logical principles of the form: if x is recognized then do y (y could be either a verbal, or textual response) [55].

Chat bots that are built on hardwired programs to look up keywords and pulling out the appropriate response for that keyword(s). The type of chat bots that I am interested in is the kind that do have artificial intelligent capabilities that give them the ability to learn and expand their knowledge base when chatting with others [13].

A good example of a chat bots is ELIZA and ALICE; both are artificially intelligent programs capable of conversing using natural language. They were built in a way that can learn from the conversations conducted with humans [54, 56, 57].

1.4.3 Using Stylometry to tell Chat bots from Humans

Studies conducted so far on separating chat bots from humans where mainly focused on chat bots' response times [58]. Other studies were looking for chat bots used as malware, these chat bots mainly keep sending URLs repeatedly in their chat [59]. Zi and Sushil also used the URL ratio; their system has four components [60]:

- The entropy component: Study time intervals between tweets.
- The machine-learning component: Study tweets contents.
- The account properties component: Study URL ratio.
- The decision maker: Based on Linear Discriminant Analysis (LDA) to decide if tweets are generated by humans or chat bots.

To the best of my knowledge, telling chat bots from humans using stylometry features was never done before. Chat bots' responses are coming from artificially intelligent knowledge base that has rules, so the chance for a chat bot to present a misspelled word is close to none, unless the chat bot programmer did add them intentionally. But in general, humans tend to misspell more often than chat bots. Also, chat bots tend to respond to the text they receive with a slight modification and

rephrasing of the original. A chat bot that can detect these techniques could be able to tell chat bots from humans and can be used as an artificial judge on a Turing test [59].

1.4.4 Features that include inter and intra-paragraph stylistic issues

Understanding sentences is not enough to interpret the complete text. Understanding relations between sentences is required, i.e. knowing the relationship structure between sentences and paragraphs of the text. Such prior knowledge is essential for applications using NLP for the purpose of text summarization or question answering [61].

Theories like Rhetorical Structure Theory (RST) [62], states that “discourse structure” can be viewed as a tree structure whose end leaves are the “elementary discourse units” (edus) of the text, e.g. sentences or clauses. Contrast or elaboration relationship can relate those edus. These relations then construct the text segments. The intermediate segments can then relate to each other to form bigger segments.

Discourse parsing is going back from the big segments to the smaller to the edus automatically. This can be achieved with the following three tasks:

- i. Identifying the edus.
- ii. Identifying the relation between edus.
- iii. Identifying the relation between segments [61].

These tasks are hard to implement for large segments. It did had some success for small structures of phrases [63, 64].

Sporleder and Lascarides claim that edus across different sentence structures cannot attach to one another. In addition, no attachment between a sentence and another preceding or following sentence. Using the “divide-and-conquer” approach, the number of valid trees will be reduced. Inter-paragraph structure can then be treated as independent from intra-paragraph structure [61].

Yaari’s experiments on merging segments were based on word co-occurrence. This was achieved by implementing a similarity measure. Sporleder and Lascarides chose a combination of clustering and machine-learned model of segment relatedness. Both results were compared against each other. Yaari uses a cosine measure to define segment similarity as shown in (Equation 2) [65, 66]:

$$\text{similarity}(S_i, S_j) = \frac{\sum_t w_{t,S_i} \cdot w_{t,S_j}}{\sqrt{\sum_t w_{t,S_i}^2 \cdot w_{t,S_j}^2}} \quad (2)$$

From (Equation 2), t ranges over the set of terms in the text. Removing closed-class words from the text then using Porter's stemmer to preprocessing the text. Based on the product of three factors, each term will be assigned a weight w_{t,S_i} , which will be assigned to term t in segment S_i . The three factors are:

- The frequency of the term in the segment f_{t,S_i} .
- The relative frequency of the term in the text $\frac{f_t}{f_{max}}$.
- The general significance of the term in a corpus ($G_{S_i g_t}$).

See (Equations 3 and 4).

$$w_{t,S_i} = f_{t,S_i} \times \frac{f_t}{f_{max}} \times G_{S_i g_t} \quad (3)$$

$$\text{Such that, } G_{S_i g_t} = \log \frac{N}{N_t} \quad (4)$$

Where N is the number of files in the corpus and N_t is number of files that contains the term t [61]. Note that the $G_{S_i g_t}$ is actually a form of the well-known IDF. That really makes the w_{t,S_i} actually a representation of the TFIDF multiplied by the f_{t,S_i} .

1.4.5 Metrics that are best for similarity rather than for exclusion of authorship

Two well-known approaches for similarities:

- **K Nearest Neighbor:** It is considered as one of the top ranked methodologies used for text categorization. Starting with a random input document, depending on the K value which determines the number of nearest surrounding documents based on similarity scores, the document will be assigned a category [66, 67].
- **Rocchio:** With a reduced overhead processing cost over KNN, Rocchio starts by assigning a centroid document to each category. Test documents will then be assigned to categories based on similarity scores with the centroid document. A noticeable reduction in processing cost over KNN can be easily achieved for larger datasets and more text documents to categorize [67, 68].

1.4.6 Metrics that are best for exclusion rather than for similarity of authorship

Two main dissimilarity metrics are:

- Minkowski Metric: Intra-dimensional and inter-dimensional weights are the main values to consider in this metric. Intra-dimensional weights are subtractive, this means, when looking for distance between vectors, the values of the co-occurring weights are subtracted as shown in (Equation 5).

$$D_{jk} = \left(\sum_{i=1}^n |x_{ij} - x_{ik}|^\gamma \right)^{\frac{1}{\gamma}} \quad (5)$$

Where $\gamma \geq 1$ and D_{jk} is the sum of the dissimilarities between x_k , the vector that is already classified, and x_j , the vector to be classified for the set of weights $i=1, \dots, n$.

When $\gamma = 1$, the algorithm is called Manhattan metric, and when $\gamma = 2$ it will be referring to Euclidian metric and if $\gamma > 2$ it represents Supermum metric.

- Canberra Metric: A major difference with Minkowski is the normalization of its calculation for each document. So the metric will do summation of the difference between co-occurring weights over the sum of co-occurring weights as seen in (Equation 6) [67].

$$D_{jk} = \sum_{i=1}^n \frac{|\vec{d}_{ij} - \vec{c}_{ik}|}{(|\vec{d}_{ij}| + |\vec{c}_{ik}|)} \quad (6)$$

Where D_{jk} is the dissimilarity score between the vector representation of document d_j and the category prototype vector C_k for $i=1, \dots, n$.

Stamatatos proposed another approach called Common N-Gram (CNG) approach. A profile is constructed for each author in the set of all authors A . The profile content is a pair of the L most frequent N -grams with their corresponding frequencies in a descending order. When a test text file (x) is required to be assigned to one of the authors (T_a), the file is sent to a dissimilarity function to measure distance as seen in (Equation 7) [30, 37, 69]:

$$d_0(P(x), P(T_a)) = \sum_{g \in P(x) \cup P(T_a)} \left(\frac{2(f_x(g) - f_{T_a}(g))}{f_x(g) + f_{T_a}(g)} \right)^2 \quad (7)$$

Where $f_x(g)$ and $f_{T_a}(g)$ are the frequencies of the n -gram g in the test text and the

author a 's training text respectively, and $f(g)=0$ if $g \notin P$.

KNN measure with $K=1$ is then used in order to predict the most-likely author of the text x in (Equation 8):

$$author(x) = arg_{a \in A} \min d_0(P(x), P(T_a)) \quad (8)$$

1.4.7 Text Classification

Text classification can be implemented based on genre or topic. Koppel et al. implemented Arabic text classification based on ideological and organization affiliation. The main four proposed ideologies were: Salafi-Jihadi, Mainstream Islam, Muslim Brotherhood, and Wahhabi. The four organizations are: (Hamas, Hizballah, Al-Qaeda, and Muslim brotherhood [70].

It is quite important for security and law enforcement agencies to be able to categorize documents in hand by ideologies in a timely fashion. With thousands and thousands of documents, it would be very time consuming to do this manually. There could be a chance that the value of a specific document is no longer important because of this delay.

Defining text features to represent the text files is the first step in text categorization. Then each document will be represented as a vector. A learning algorithm can be used to distinguish between these vectors [15].

Two main issues to consider ahead of time are the selection of features, and the selection of the learning algorithm. When classifying for topics, features to be selected must reflect content, while if we classify for writing style we select features that reflect style. Function words can be used to represent style, but they are very useless for content or topic based classification. The type of problem under hand can be a key element in choosing the feature selection [70].

The work of Abbasi and Chen in contrast to Koppel's work was to identify authors posting from extremist sites, while Koppel's was to identify organizations and ideologies [21].

The most 1000 frequent words were selected as the feature, with no stemming since stemming is so time consuming in Arabic language. The function words were not excluded from the 1000 frequent words. Bayesian Multi-class Regression (BMR) was used for learning. Each feature is assigned a weight so one can make sure to

highlight the most important features for each category. Ten-fold cross validation was used for testing. This can be accomplished by dividing corpus into 10 segments, learn with 9 segments and test with the tenth. Shuffle data and repeat process with new learning and testing sets [70].

Koppel et al. demonstrated that using stylistic and content features were successful for automatically identifying documents to its organization or ideological group, and the robustness of this technique makes it reliable for the law enforcement purposes [70].

Koppel and Schler tried to solve the authorship verification problem by “testing the rate of degradation of the accuracy of learned models as the best features are iteratively dropped from the learning process” [71].

It would be hard to collect data for Shakespeare versus not-Shakespeare and classify from this perspective directly. How would it be possible to define all non-Shakespeare samples? What will be reasonably enough sample size covering this class? In addition to that, how can we avoid author consciously changing his/her writing or style drifting over time? Researchers must learn “how to distinguish between shallow differences that reflect conscious or non-conscious changes from same author versus deep changes reflecting other author” [71].

Two approaches have been experimented with:

- A. Naïve approach for authorship verification: when studying the work of author A against the work of author X, one should begin with splitting the text into chunks. Using k-fold cross-validation, one could easily distinguish between A and X. Consequently, one can conclude that A is not the author of X. Although this approach yielded an accuracy above 98% for concluding that none of the three novelists (Herman Melville, James Fenimore Cooper and Nathaniel Hawthorne) wrote *The House of Seven Gables*, but Hawthorne actually was the author [71].
- B. Unmasking: The idea behind unmasking can be simplified as follows, if we have two samples for the same author, but a small set of features does disrupt

the similarity, unmasking will remove these features and the similarity will be obvious [71].

Figure 4: Unmasking "An Ideal Husband" against ten authors

When applying the k-fold cross-validation, Figure 4 shows that one author tends to be separated with each feature elimination process, and all other authors are in one group, this means that this separated author is the actual author [38, 71].

1.5 Authorship versus Text Similarity

When looking for authorial traits, one is looking for stylistic similarities between documents for the same author. In contrast, when needing to measure similarities between two texts, firstly, these texts should be represented as a vector using some feature representation, and then a similarity measure, like the cosine similarity, can be applied. Equation 9 gives the cosine similarity equation used.

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (9)$$

A and B are the vectors representing the two documents, the smaller the angle θ the higher the similarity between the two texts as shown in Figure 5 below [72].

Figure 5: Cosine similarity using vector representation

Text similarity can be measured on a document level between documents, and can be implemented on a paragraph level between paragraphs within the same document or between paragraphs of different documents.

Plagiarism detection is actually straightforward to implement, a 100% similarity is a complete match and hence plagiarism.

1.6 Quality of Writing (QoW) as an IQ Measure

Do authors with high Intelligence Quotient (IQ) have a better quality contents in their writing? If yes, is there a correlation between the two factors? This is an interesting topic, and there might be a correlation that could be measured. Firstly, to think about measuring the quality of a text, one should come out with a feature(s) that will successfully reflect this measure. One can think of vocabulary richness as one of the features, the frequency of long words as another feature.

Obtaining the IQ score for candidate authors is a real challenge. This data is not publically available; it is available online only for limited number of celebrities, and obtaining such data was close to impossible. A simplification has been proposed to get around this obstacle by looking at the problem as a binary classification problem. To comparing between authors who are known to have a high IQ, i.e. scientists and samples from grade school students.

1.7 Conclusion

The plethora of online text in various forms, from books, novels, published papers, blogs, emails, and even short chat messages, increase the need for further

work and research in authorship attribution. Various features and methodologies have been experimented. For each different type of text, different features and methodologies will be required to obtain the best results.

When doing text classification or categorization, we should clean the text from function words, since these words will disrupt the word frequency distribution and yield a misleading classification results. If dealing with a closed set of possible authors, we can use text classification or categorization techniques as a way to identify authors, but if number of authors is unlimited then different methodologies need to be investigated. Function words or stop words can be used as an identifying feature in the case of authorship identification. Various authors tend to use function words differently, so the use/elimination of function words basically relies on the goal of the analysis. If classifying then we need to clean the text from function words, and if one is looking for authorial traits then function words should be used.

There is a fine line between text similarity and authorial traits, text similarity will measure how much two texts are similar and authorial traits will measure if a text is for one author depending on a similarities with another text for that same author.

More features can be used to measure quality of writing (QoW), and a correlation could be investigated to one's IQ.

CHAPTER II

PRELIMINARY WORK

2.1 Chat Bots

“A Chat bot, Chatter bot, Chatter box or Chatter robot is a computer application designed to simulate a conversation with a human” [48]. Chat bots are mainly used in applications when human can be replaced with a bot to serve customers, such as online help, e-commerce, customer services, call centers, and internet gaming [49].

Chat bots are typically perceived as engaging software entities, which humans can talk to. Some Chatter bots use sophisticated Natural Language Processing Systems (NLPS), but another type will just scan for keywords within the input and pull a reply with the most matching keywords [50]. Chat bots are still a largely developing technology; consequently, quality of simulated conversations varies from realistic to mostly nonsense [51].

2.2 Turing Test

The Turing test is a test used to determine a machine’s ability to illustrate intelligent behavior, in a manner difficult to distinguish from that of an actual human [73]. This test was introduced by Alan Turing, which opens with the words: “I propose to consider the question, Can machines think?” [74].

An example of this is the Loebner prize, in which a human judge engages in a natural language conversation with humans and machines designed to generate text in a way indistinguishable from human being [75]. If the judge cannot reliably

determine the machine from human, then that machine is said to have passed the Turing test.

The test does not check for correct answers from a machine; instead it checks for how closely the answers resemble a typical human response.

2.3 Motivation

Individuals committing online criminal activities utilize chat bots as a new way to steal private information and commit fraud and identity theft. The need for identifying chat bots by their style is becoming essential to overcome the danger of online criminal activities.

Researchers realized the need to advance the understanding of chat bot and design programs to prevent criminal activities, whether it is an identity theft or even a terrorist threat. The more research developers conduct to advance their chat bot ability to perceive humans, the more duties need to be followed to confront those threats by the research community; which means, more intelligent chat bots will require a more complex algorithms to identify them [53].

To get authorship attribution techniques publically accepted and implemented, it has to meet the Frye [76] and the Daubert [77] tests. Frye test states that: “scientific evidence is admissible only if generally accepted by the relevant scholarly community”. While the Daubert tests is more sophisticated and incorporates several more objective tests, including but not limited to empirical validation of the science and the techniques used [78].

Currently, authorship attribution does not meet the above-mentioned criteria. Authorship identification techniques could not pass the Frye test, and the lack of understanding of practices and known error rates prevented those techniques from passing Daubert consideration too [78].

2.4 Data Collection

Hindi developed a C# application to connect two chat bots from a list of online available chat bots and start a chatting session between them for some time [79]. The application saved three text files for each conversation session: one for the complete session, one for the conversation from the first chat bot engaged in the conversation, and one for the conversation from the second chat bot.

Figure 6 represents the flow chart of the application used for collecting the data. To obtain more data, chat logs from the chat bots' websites were used. These logs were usually a chat logs with human users for the purposes of training the bot to gain more human-like responses.

A reasonable amount of data was collected; un-necessary meta data was removed, such as the chat bot name and time and date of conversation. Some logs were repeating same statements over and over from that bot; these logs were discarded.

Figure 6: The Chat bot Reloaded Flow Chart

For each bot, all of the logs were merged into one text file. Then a free ware file split was used to split the one text file into individual 4KB text files. The reason behind this was to normalize the data set when testing; making sure that the same amount of data was used for the experiments. Figure 7 shows a sample text from a conversation between two chat bots

Jeeney: hi. How are you today? SkyNet: I am excellent this refreshing winter day. Jeeney: What do you think of me? SkyNet: I have not come to a conclusion about what of me. Jeeney: Why didn't you? SkyNet: I thought it was the right course of action. Jeeney: Why did you think that? SkyNet: It is part of who I am. Jeeney: Oh, okay.	TalkBot: Hi, I am TalkBot. Hal: I have never heard of that name. What nationality is TalkBot? TalkBot: Not that I recall. I often hear many things from people who drop by to chat but I sure don't remember it all. Hal: Not many people express themselves that way. TalkBot: Perhaps, but a significant number nevertheless. Hal: Sounds okay.
--	--

Figure 7: Sample Conversation between Chat bots

2.5 Applications Used for Testing

Initially, we had no prior knowledge as whether chat bots had a style to apply the authorship identification techniques or not, so before going further in that direction, data needed to be tested by available freeware software to determine whether we could study the data for style. Two applications were used for this purpose:

a. Stylometry Project

This is a project by PACE University, mainly built for emails authorship attribution [80]. Initial experiments on the data did not show any promising results, so no further experiments were conducted using this application.

b. Java Graphical Authorship Attribution Project (JGAAP)

This is a project performed by a Patrick Juola, a scholar in the field of authorship attribution [17]. JGAAP has an interface and flexible options to choose from in each stage of the application.

JGAAP consists of five stages:

1. Documents: In this stage a user can:
 - a. Load the documents
 - b. Define the documents that have an author
 - c. Define the documents that have unknown author.Figure 8 shows a snapshot of this stage.
2. Canonicize: User can select the preprocessing that needs to apply on the documents, a user can select one or more of these preprocessing or can select Null Canonicizer as shown in Figure 9, examples on Canonicizers are:
 - a. Normalize white space: Take out extra spaces from the text.
 - b. Strip Alphanumeric: Allowing only alphabetic and numbers to show in the text, no symbols.
 - c. Strip Punctuation: To subtract the punctuations from text.
 - d. Unify Case: Change the case for all letters to lower case.
 - e. Null Canonicizer: No preprocessing, just take the text as is without any type of preprocessing.
3. Event Set: the user can select the feature needed to extract from the text; the only drawback in this stage is that the user can only select one feature at a time. Table 1 shows all these feature selection option with their description, and Figure 10 shows this stage in JGAAP.
4. Analyze: Choosing the classification method that will be used to classify the data set. A wide set of classifiers are available in this stage as shown in Figure 11.

5. Report: Will output a text for all the text with unknown author with their predicted author. Figure 12 shows an example of this stage.

Figure 8: Documents loading stage in JGAAP

Figure 9: Canonicization stage in JGAAP

Figure 10: Event Set or feature selection stage in JGAAP

Figure 11: Analyze or Classification stage in JGAAP

Figure 12: Report Stage in JGAAP

2.6 Chat bots used

Initial data were collected for eleven Chat bots, but only six Chat bots were used in the experiments based on data collected volume. Those chat bots were:

- Alice: Artificial Linguistic Internet Computer Entity created in the Artificial Intelligent Markup Language (AIML). It was developed by Dr. Richard Wallace, who graduated from Carnegie Mellon University (1989) [57].
- CleverBot: Created by Rollo Carpenter [81].
- Hal: Created by AI research Facility as a virtual child capable of acquiring information like a child [82].
- Jeeney: An artificially intelligent chat agent designed to learn from fluent English interaction [83].
- SkyNet: An online artificial intelligent. It specializes as an intelligent agent, answering questions and acting as a web portal. Developed by Ken Hurtubies [84].

- TalkBot: Another live agent developed by Wendell Cowart that crashed land on earth on January 2, 2001 [85]. It is the three time winner of the Chatterbox challenge and finished second in the 2002 Loebner Contest [86].

As mentioned previously, chat bot Reloaded performed data collection. Some chat bots did not generate a quality logs when chosen by chat bot Reloaded, thus, logs from these chat bots' websites were downloaded and used in the experiments. Figure 13 shows a snapshot of the Chat bot used taken from their websites.

2.7 Behavioral Drift

Writing is a skill, and like any other skill, authors will gain new knowledge and new expertise with time. These will lead the authors' style to evolve and change, a concept known as Behavioral Drift [87].

Studying the Chat bots from this aspect was very interesting, since chat bots are essentially a computer programs, built upon artificially intelligence algorithms.

The study by Ali et al. did show a sign of behavioral drift for some chat bots, but mainly the study did not confirm the results due to the shortage of chronological data [53].

Data was mainly collected from Loebner prize's website [75], but since only the top three winners' logs are available, data for the same chat bot for different years was hard to obtain, unless that chat bot was a winner for several years.

Figure 13: Chat bots used in the research taken from Chat bots' website

2.8 Content Analyses and the Other IQ

Content analysis is a broad set of methods for inferring psychological variables from documents [88]. The classic Thematic Apperception Test [89] has been used to assess political speeches and other communications on power, achievements, and affiliation motives [90]. The goal in this part of the research is to study text content looking for a correlation between quality of contents and the Intelligence Quotient (IQ) [91].

2.9 Term Frequency Inverse Document Frequency (TFIDF) measure

TFIDF is one of the most widely used features in Stylometry; it is a way to assign a value to each token representing a weight for that token. For example, a word

that occurs in each and every document in the corpus will have no value classifying these documents. On the other hand, a token occurring in a smaller set of documents will have a higher value that will help in corpus classification.

In Information Retrieval (IR) domain, documents are represented in vector space model. After tokenizing and stemming these documents, each token is represented as an axis in the Euclidean space. These documents are vectors in this n-dimensional space. For each token (term) in a document (d), where there are (N) documents, we can define the Inverse Document Frequency (IDF) as:

$$IDF = 1 + \log\left(\frac{n_t}{N}\right) \quad (10)$$

The term (n_t / N) in (Equation 10) represents the rarity of the term (t), such that (n_t) is number of occurrences for term (t) in document (n) over occurrence of (t) over all documents. This rarity measure is also considered as an importance score for that term [92].

Other form of the TFIDF will have different IDF calculation as see in (Equation 11).

$$IDF = \log\left(\frac{\|N\|}{1+n_t}\right) \quad (11)$$

We added 1 to the denominator to avoid division by zero when the term frequency for that document was zero.

Term Frequency (TF) is another measure calculates the number of times term (t) occurred in document (d) relative to the total number of terms in that document as shown in (Equation 12).

$$TF = \frac{freq(t, d_t)}{\|d_t\|} \quad (12)$$

Such that $freq(t, d_t)$ will calculate the frequency of term (t) in document (d_t), and the $\|d_t\|$ is the total number of terms (tokens) in document (d_t) [92].

So the TFIDF will be as followed in (Equation 13).

$$TFIDF = TF * IDF \quad (13)$$

So what does the IDF means? Assuming that a term (t) appears in all the documents of the corpus, which will leads to the values of $\|N\|$ and (n_t) to be the same, and the log will be zero, and the IDF will be 1 from (Equation 10), and the TFIDF will equal to TF in this case, and a value close to zero in (Equation 11).

The closer the TFIDF value get closer to zero, the less weight that term will have to classify that document.

2.10 Byte-Level N-Gram Term Frequency Inverse Token Frequency

The idea behind this feature came from the increased importance of the N-Gram feature and the TFIDF measure in classification tasks. Several research studies showed increased accuracy when using either one of those two features [30, 33, 36, 93-95]. Treating the text as characters rather than tokens as explained in Figure 2, the Byte-Level N-Gram slide over the characters and form the N-Grams depending on the value of N, (See Appendix A) [96].

BLN-Gram-TF-ITF will implement the idea of TFIDF but with different aspect, in this case the document will be the token, and the token will be the term generated from the N-Gram, (See Appendix B).

Assuming that the following text saved in List1:

List1="I will try to check my feature using this text as an example"

List1 will be transferred to lower case, so the new list will be:

List2="i will try to check my feature using this text as an example"

If N=3 is selected, then the N-Gram of this list1 will be:

N-gram-List= ['i w', ' wi', 'wil', 'ill', 'll ', 'l t', ' tr', 'try', 'ry ', 'y t', ' to', 'to ', 'o c', ' ch', 'che', 'hec', 'eck', 'ck ', 'k m', ' my', 'my ', 'y f', ' fe', 'fea', 'eat', 'atu', 'tur', 'ure', 're ', 'e u', ' us', 'usi', 'sin', 'ing', 'ng ', 'g t', ' th', 'thi', 'his', 'is ', 's t', ' te', 'tex', 'ext', 'xt ', 't a', ' as', 'as ', 's a', ' an', 'an ', 'n e', ' ex', 'exa', 'xam', 'amp', 'mpl', 'ple']

For each unique term in N-Gram list, the TF-ITF will be calculated as followed:

$$TF = \frac{freq(t, N-Gram)}{\|N-Gram\|} \dots \dots \dots (14)$$

In which the calculated frequency will be based on the times this term generated from the N-Gram occurred in the N-Gram list divided over the length of N-Gram list, which also represents how many terms, this list have.

The ITF will be calculated as followed:

$$IDF = \log\left(\frac{\|NT\|}{n_{T \text{ with } t}}\right) \dots\dots\dots(15)$$

Where $\|NT\|$ is the total number of tokens in the corpus, and the $(n_{T \text{ with } t})$ is how many tokens (T) containing the term (t). The number 1 was not added to the denominator because the calculation for the frequency of Tokens having term (t) did that already if the frequency is equal to zero, (See Appendix B).

The logarithm function used in this experiments was the natural logarithm with base $e = 2.302585092994046$.

For each document, the same process will be done and all the terms with their corresponding TFITF value will be saved to a Comma Separated Version (CSV) file to be exported later to Matlab[®].

For example, the $tf('ry', N\text{-Gram-List})$ will be $2/58 = 0.03448275862069$

Then calculating the ITF for that, and multiply both results to get the TF-ITF

$ITF = itf('ry', N\text{-gram-List}) = 4.0943445622221$

$TFITF = TF * ITF = 0.14118429524903794$

Figure 14 describes the flow chart of the algorithm used to create the TF-ITF.

The TF-ITF is calculated for each term generated by the $(byte_N_gram)$ method, See Appendix A.

Another version was created by eliminating spaces from the feature creation process, and by using variations of unique tokens versus all tokens per file when creating the lists.

Figure 14: Flow chart of the Byte-N-Gram-TF-ITF

2.11 Conclusion

Studying chat bots for stylistic traits is a novel approach. Previous researches attempted to separate humans from bots, mainly by response time statistics, since bots tend to respond faster and with a regular timing pattern [97, 98]. Initial experimentation demonstrates that chat bots do have a style and one can successfully identify the correct chat bot behind a text. Byte Level N-Gram Term Frequency Inverse Token Frequency (BLN-Gram-TF-ITF) was created and tested on chat bot text and human corpus from Gutenberg project [99]. Additional versions of the feature and additional features have been combined with the proposed feature and have been tested against human corpus.

CHAPTER III

EXPERIMENTS AND RESULT ANALYSIS

3.1 Detecting Authorship

Initially, we had no prior knowledge as to whether the data collected for chat bots had any kind of style. Thus, it was easier to pick software with successful results on human corpora to do the test. Java Graphical Authorship Attribution Project (JGAAP) is an open source freeware produced by Patrick Juola [17], with the option of choosing different

The experiments started by loading the documents for the six chat bots under study, the authors in this case. For each preprocessing, various selections of event set and classification algorithms were chosen and the outcomes recorded for each run. Figure 15 shows how the experiments were performed preprocessing, features, and classifiers on the data set.

Figure 15: Flow Chart of the process followed during the Experiments

A total of 306 different tests were conducted on the data set. Figure 16 and Figure 17 show average accuracy for each classifier used and average accuracy for each feature used respectively, and it shows how the data interacts with different selection of features and classification method.

Figure 16: Average accuracy for each classifier

Figure 17: Average accuracy for each feature

The Juola & Wyner Cross Entropy classifier achieved the maximum accuracy (72.05%) with a drawback of slow performance [52].

Table 7: Average accuracy for each feature over all classifiers

Feature Used	Average per Feature	Feature Used	Average per Feature
Vowels 2-3 letters words	60.30%	MW function Words	52.46%
Vowels 2-4 letters words	59.48%	Word BiGrams	51.87%
2-4 Letters	59.13%	Vowels 3-4 letters word	49.77%
Vowel initial words	58.31%	Word Length	39.58%
2-3 Letters	58.20%	Word TriGrams	38.64%
Character BiGrams	57.26%	Syllables per word	28.57%
Characters	56.56%	Hapax-Dis Legomena	24.36%
Character TriGrams	55.74%	Word Tetra Grams	24.36%
Words	55.39%	Hapax Legomena	22.83%
Character Tetra Grams	55.27%	Dis Legomena	20.96%
3-4 Letters	53.51%		

Table 7 shows that maximum accuracy (60.30%) was achieved using the Vowels 2-3 words feature, which corresponds to the words starting with a vowel and with length of 2 to 3 characters.

Based on the data analysis of the results obtained from these experiments, we concluded that chat bots do have a style, and they could be identified by their style of writing.

3.2 Detecting Behavioral Drift

3.2.1 Data Collection and Preparation

For the purpose of behavioral drift study, the data should be collected over several years. For this reason, Loebner prize website was selected, since it has the winners' logs for different years [75]. The problem was choosing the chat bots that had won more than one year, and having the current data to compare against. The main idea was to get data for a chat bot from different set of years and study for style change by comparing these styles with the latest style data available for that specific chat bot. Figure 18 shows a sample conversation between the judge and a chat bot, Alice. The chat bot was labeled as program, so there is no prior knowledge from the judge about whether a chat bot or a human was on the other ends. Figure 19 shows the chat log after meta-data cleaning.

After collecting the needed data, a Perl script was used to separate the chat logs into judge text file and a chat bot text file, and to remove all meta-data like the date, time and name of chatter. Only three chat bots were eligible for this study, since they were winners for several years and their latest 2011 data was available. These chat bots were Alice [57], Jabberwacky [100] and Jabberwock [101].

Copyright (C) 2003 Cambridge Center for Behavioral Studies. All rights reserved.
ALICE Silver Edition conversing with Judge 06

2003-10-18-15-40-46 PROGRAM: Hello? Is anyone there?
2003-10-18-15-41-00 JUDGE: Hi. who are you?
2003-10-18-15-41-05 PROGRAM: Hello there. I am Alice. who are you?
2003-10-18-15-41-22 JUDGE: I'm Marilyn. what are you doing here?
2003-10-18-15-41-27 PROGRAM: Hi there Marilyn. I am chatting with people online.
what are you doing?
2003-10-18-15-41-48 JUDGE: well at the moment I'm talking to you Alice?
2003-10-18-15-41-53 PROGRAM: well that's okay. Sounds like fun.
2003-10-18-15-42-11 JUDGE: It could be. what shall we chat about then?
2003-10-18-15-42-16 PROGRAM: Under what circumstances? Quite a lot of things.
2003-10-18-15-42-51 JUDGE: As we have only just met perhaps we should just get to know each other
2003-10-18-15-42-56 PROGRAM: Do you mean your name is we have only just met perhaps we should just get to know each other?
2003-10-18-15-43-15 JUDGE: No. My name is Marilyn
2003-10-18-15-43-20 PROGRAM: what is your real name? I am always glad to make new friends, Marilyn.
2003-10-18-15-43-43 JUDGE: That is my real name. Are you really called Alice?
2003-10-18-15-43-48 PROGRAM: who gave you your real name? I don't know whether or not I am called.
I am a Mammal.. Really.
2003-10-18-15-44-19 JUDGE: My parents gave me my name.
2003-10-18-15-44-24 PROGRAM: Such an original name!
2003-10-18-15-44-42 JUDGE: Thank you. who are you called after?
2003-10-18-15-44-47 PROGRAM: you are quite welcome. Human beings. I have never been asked that before. How did you hear about Alice?
2003-10-18-15-45-08 JUDGE: You said your name is Alice
2003-10-18-15-45-13 PROGRAM: ALICE is my name too!
2003-10-18-15-45-29 JUDGE: we are going round in circles
2003-10-18-15-45-34 PROGRAM: Do you think I am going round in circles too?
2003-10-18-15-45-53 JUDGE: Yes. Shall we talk about something else?
2003-10-18-15-45-58 PROGRAM: why? Sure we can talk about whatever you want.
2003-10-18-15-46-13 JUDGE: where are you Alice?
2003-10-18-15-46-18 PROGRAM: I am in Dr. Richard S. wallace's home in oakland, CA.

Figure 18: Raw chat data before cleaning from the Loebner Contest

Hello there. Searching...Searching...Please stand by. What color are your eyes?
That is a very unusual color. You can design bodies for me with any color.
Black.
I'm glad your impressed, JUDGE 1. Triangular.
Makes sense to me. Check back later and see if I learn the answer to that one. Not many people express themselves that way.
You have reached the limits of my knowledge. That's not something I get asked all the time. What do you really want to ask me?
I am in Dr. Richard S. Wallace's home in Oakland, CA. Where are you?
Which part of England? I don't think he wants me to tell you.
What is it like in there? Only on the web.
The first one, I think.
Sorry I can't see annything right now. My eye is off.
I have one master.
What does "that" refer to?
Thank you for your frankness.
Yikes! Do you make a lot of sacrafices for others? I had one eye but it's not attached now.
If I have to choose, I pick never had it from birth.
You have reached the limits of my knowledge. I would do a search for it. Have you ever been to Europe?
Which country do you live in? Once or twice.
When? Can you rephrase the question please?
Is that your final answer? Which part of England. I am living in Oakland, CA. Where are you?
I have no idea where that is. Tell me a little about your life in Final answer. Do you feel more body sensations than emotions?
Cool. I have a vocabulary of 10000 words.

Figure 19: Chat bot Sample chat log after separation and cleaning

3.2.2 Behavioral Drift Experiments

All experiments for this part of the research were conducted using RapidMiner[®] [102]. RapidMiner[®] (RM) is an open source free ware and mostly Java based. It allows doing almost all Data Mining tasks through a Graphical User Interface (GUI), such as drag and drop environment, selecting your data source, and building your model, and running your data for results. Figure 20 represents the model used for training the data, Figure 21 shows the preprocessing stage, while Figure 23 shows the testing model used.

Figure 20: The Model used for Training in RM

The preprocessing stage is a set of steps performed to allow statistics to be used on tokens. First, we unify the letter case to lower case so “Tree” and “tree” will be one token. After that, we tokenize the text, using the white space in the text. Stemming takes each token to its root after stripping all the prefixes and suffixes, if any. For example, “fishing”, “fisher”, “fishes” will all have the root “fish”.

N-Gram is using the sliding window technique to get consecutive tokens together depending on N value. So if $N=3$, we take each three consecutive tokens as a set of N-Gram. See Figure 1 for more details.

The output of the Process Document will be a Word List, or Bag Of Word (BOW). Figure 22 shows a snapshot of the word list, which is a list of all tokens in the documents and number of occurrences for each of the documents.

Figure 21: Preprocessing Stage

Word	Total Occur...	Document O...	alice	cleverbot	alan	Hal	Jabberwock	Jeeneey	MyBot	SkyNet	Suzette	TalkBot	JABBERWACKY
access	6	6	0	0	0	1	0	0	0	5	0	0	0
accord	9	8	0	1	3	0	0	1	0	4	0	0	0
act	12	9	0	1	2	0	1	0	2	0	6	0	0
action	7	7	2	0	1	1	0	0	1	0	2	0	0
activ	17	16	3	0	3	1	1	0	1	2	6	0	0
actual	30	25	2	3	1	2	4	8	0	0	3	4	3
admit	6	6	0	2	0	1	0	1	0	0	1	0	1
afraid	24	18	0	2	1	3	0	4	0	2	9	0	3
ag	10	10	1	2	1	2	0	0	0	2	1	0	1
agre	59	34	7	0	13	7	3	0	14	3	1	7	4
ahead	9	8	1	2	1	2	1	0	0	1	0	1	0
ahh	13	9	0	0	0	0	0	0	2	11	0	0	0
alan	17	9	1	0	12	0	0	0	3	0	0	0	1
alic	28	11	12	3	1	12	0	0	0	0	0	0	0
alien	13	7	0	2	0	0	3	0	8	0	0	0	0
alik	11	11	0	1	2	4	0	0	2	0	0	0	2
allow	7	7	1	0	0	1	0	3	0	0	0	0	2
alon	19	16	0	3	0	2	2	0	0	0	1	7	2
alreadi	59	40	4	5	10	1	18	3	7	1	5	0	5
alright	32	23	4	1	7	0	3	15	1	0	0	1	0
alrighti	7	6	0	0	0	0	0	0	0	0	0	7	0
alwai	60	42	16	4	3	6	6	5	4	2	3	8	3
amaz	12	12	0	0	1	0	1	2	0	1	1	5	1
amus	11	10	3	0	2	0	1	0	2	3	0	0	0
angri	9	7	0	1	0	2	0	0	0	3	0	0	3
ani	107	57	4	14	4	9	4	28	3	6	20	6	9
anim	13	7	0	2	0	3	0	2	5	1	0	0	0
annoi	9	6	0	1	0	0	1	0	0	0	3	1	3
anoth	43	34	2	2	4	2	8	1	8	6	2	7	1
answer	102	55	7	19	8	10	9	3	13	10	2	4	17
anybodi	12	9	0	0	0	4	2	3	0	0	0	3	0
anymor	8	8	0	2	0	0	2	0	0	0	0	0	4
anyon	21	18	0	5	0	1	4	0	0	0	8	1	2
anyth	96	55	10	11	3	4	6	12	3	10	4	20	13
anywai	12	11	0	0	2	0	1	0	3	0	0	4	2
anywher	7	7	0	2	0	2	0	1	0	0	1	1	0
apolog	12	9	3	1	0	0	1	0	0	0	2	2	3
appar	9	7	0	1	0	3	0	4	0	0	0	0	1
appeal	10	8	1	0	1	0	0	7	0	0	0	0	1
appl	16	8	0	1	8	0	0	0	0	0	4	3	0
appreci	16	12	0	0	2	0	9	0	0	4	0	1	0
approach	7	7	0	0	0	1	1	2	3	0	0	0	0

Figure 22: Wordlist or Bag Of Word (BOW) saved

Figure 23: Testing Model used in RM

The experiments were conducted on two different sets of reference data and saved in two different sets of Word List and Models, first set using all eleven chat

bots' data collected in 2011 for training and the other set of data was using only the three chat bots under study data from 2011 for training [53].

Table 8: Alice's Confidence Levels

	Accuracy Confidence										
	2001	2002				2004	2005				
Alice	23.9	13.1	10.5	14.6	20.2	16.5	11.2	9.2	7	7.8	11.7
Hal	12.8	10.6	14.5	12.4	14.3	11.6	11.5	12.2	12.5	12.9	14.1
Jabberwacky	7.9	8	7.2	11.3	8.6	9.4	10.1	13.5	8.4	9	9.1
Alan	7.8	12.3	10.1	7.9	9.3	8.6	9	8.6	8.9	10.7	9.4
Suzette	6.2	7.5	7.5	6.8	7.1	8.3	10.1	10.4	10.5	10.4	10.1
SkyNet	5.9	9.4	6	5.1	4.5	6.6	6.9	6.4	7.1	6	5.9
MyBot	6.4	8.2	8.7	7.1	7.3	6.8	7.5	7.8	10.7	8.8	8.4
CleverBot	11.8	12.3	11.4	11.9	10.3	12.8	13.5	12.1	11.6	11.7	12.2
TalkBot	6	7.1	7.9	8.2	6.8	7	6.6	6.1	7.5	8.6	6.1
Jeeney	6.1	6.6	9.4	8.6	6.2	7.7	6.9	6.2	8.2	6.7	5.7
Jabberwock	5.2	4.9	6.8	6.2	5.3	5	6.7	7.5	7.4	7.4	7.2

Th

e output of the experiments was the accuracy confidence levels for each chat bot tested. Table 8 shows Alice's prediction when using all the chat bots for training. The bolded and yellow shaded cells are the ones having the highest prediction value. For example in 2001, Alice was correctly predicted with (23.9) which is the highest value among that column from all chat bots values. In 2005, CleverBot has the highest prediction value of (13.5) instead of Alice's (11.2) value. Notice that adding the predictions for each file should add up to 100.

Table 9 represents the outcomes when conducting the same experiments but when using only the three bots under study for data training.

Table 9: Alice Confidence Levels with three bots training set

	2001	2003				2004	2005				
Alice	57.9	46.9	42.7	46.1	51.9	53.3	31.5	32.9	31.3	34	35.8
Jabberwacky	23.1	31.7	31.3	29	27.8	27.8	35.8	35.9	33.2	36.2	33.1
Jabberwock	19	21.4	26.1	24.8	20.2	19	32.7	31.5	35.6	29.9	31.1

Table 10 represents the prediction values for Jabberwacky when trained with all eleven chat bots. The reason there is only one entry for 2002 and three for 2003 is that the log file in 2003 was big enough to split into three files, while in 2002 the file was small, and was enough for only one file.

Table 10: Jabberwacky's Prediction values when training with all bots

	2002	2003			2004	2005				
Hal	12.9	8.5	11	8.1	9.7	13	11	12	7.2	9.5
Jabberwacky	9	18	17	21	15.6	9	10	14	18	21
Alan	10.4	8.1	7.8	7.1	8.2	11	10	11	8	6.3
Suzette	10.4	8.4	9.5	6.5	9.8	10	11	7.4	9	9.9
SkyNet	6	5.6	5.2	5.2	5.8	6	6.9	5.3	6.2	5.2
MyBot	8.8	9	10	6.9	10.9	8.8	12	11	14	7.2
CleverBot	11.7	16	15	17	15	12	15	14	15	17
Alice	7.8	6.3	4.6	7.8	6	7.8	6.3	5.3	5.7	6
TalkBot	8.6	6.3	6.7	7.9	8.2	8.6	5.9	8	6.9	8.7
Jeeney	6.7	5.8	5.2	7.1	5.1	6.7	4.8	6	5.1	4.9
Jabberwock	7.4	7	8	6.1	5.8	7.4	7.8	6	5.5	5

Table 11 shows Jabberwock's predictions when training with all eleven chat bots. Jabberwock was not identified correctly in 2002 and was one out of five correctly identified in 2005.

Insufficient chronological data leads to less accuracy than expected. Different numbers of files for different years under study will also give misleading results.

Table 11: Jabberwock's Confidence values training with all bots

	2002	2005				
Hal	8	9.8	9.8	8.9	10	11
Jabberwacky	15.6	11	8.3	12	9.2	10
Alan	8.5	11	9.2	10	8.6	8.5
Suzette	7.4	7.5	11	8.8	6.3	7.5
SkyNet	6.5	6.2	5	5.2	5.7	6.6
MyBot	7.3	8.9	7.8	6.7	9.1	8.1
CleverBot	13.5	13	10	11	12	10
Alice	10.3	8.6	5.9	7.1	10	8.1
TalkBot	7	7.6	7.8	7.2	7.8	9.4
Jeeney	8.8	9.1	6	12	10	11
Jabberwock	7.1	7.8	18	10	11	9.5

Figure 24 graphs a comparison of the three bots under study for the years that data was available. Training the model with the 2011 data only and testing with data that had never been trained gave us an indication about how the style of the bot changed over years. Another aspect that could be followed was training the model on old data and testing the newer data, however, this path was not followed due to insufficient data.

Figure 24: Prediction level over years under study

Data was not available for Alice for the year 2002 and for Jabberwock for the year 2005.

This data does show a sign of behavioral drift. However, the data used for these experiments were insufficient to give a firm and confident results, thus is not 100% affirmative. Different chat bots did show different responses, mainly due to the different learning algorithms used for different bots. Data for the years (2006-2010)

will give a better understanding of the style drifting. Unfortunately, data was not available for those years in Loebner prize.

3.3 Byte-Level N-Gram Term Frequency Inverse Token Frequency

3.3.1 Data used

The main goal of this part of the research is to compare results from experiments conducted using the JGAAP with experiments on the same data set using BLN-Gram-TF-ITF. 10 files for each of the six authors under study, each text file with size 4 Kbyte.

This section of the research was performed using Python™ [103], and the Natural Language Toolkit (NLTK) library was used for processing the text [104]. For each file in the data set, the file was tokenized and transformed to lower case. Punctuations were extracted out, and a data structure list was created to save the tokens for each file and to append tokens to this list. After obtaining all the tokens from all files, a list for the unique tokens was created and unique tokens were saved in this list.

Finally, each file was sent to the (byte-N-Gram) method to get the byte-N-Gram terms, with N equals three in this case, or what is known as trigrams (See Appendix A).

Dynamic lists were created, a list for each file having the generated terms for that file. A dynamic multi-dimensional array containing the set of features calculated by the (tf_itf) method was created and the features for each file of the 60 files were saved (See Appendix C).

The files were named in a way that each 10 consecutive files in the corpus belong to one author. The feature array contained 60 rows; each 10 rows belonging to one of the authors, and for each author, 10 rows were saved in one csv file, (See Appendix C).

3.3.2 Matlab Classification

Upon completion of the Python code, the output was six csv files saved on local machine. These files were used in Matlab code that will classify them using the KNN with N value equals to 4 and K-Fold Cross validation with K=5. By using the K-Fold cross validation, the data was split into 80% training and 20% testing. The data were shuffled, and the process was repeated five times in each run of the code, which provided an insight for outcomes over different training set with best accuracy outcomes, (See Appendix D).

As explained in Appendix D, the first step of the Matlab code was reading the csv files for each author under study then calling the (crossvalind) that would return the set of indices that would be used for splitting the data set into Training and Testing. It would then shuffle those sets for each run to give new splitting points for the Training and Testing sets.

The set of indices determined the training set, training labels, testing set, and the testing labels. At this point, testing labels were the classes or the authors' names. The knnclassify function was used for classifying these sets, and will output the prediction classes for each test file.

3.3.3 Classification Results

The confusion matrix, which is the sum of diagonal divided over the sum of the whole matrix, was used to calculate the total accuracy for our model as seen in Figure 25.

The accuracies resulted for each cross validation was as follows:

Accuracy = [91.67, 75.00, 83.33, 83.33, 83.33]

The five confusion matrices were:

$$\begin{aligned}
CM1 &= \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} & CM2 &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \\
CM3 &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 \end{bmatrix} & CM4 &= \begin{bmatrix} 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \\
CM5 &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 \end{bmatrix}
\end{aligned}$$

Figure 25: Confusion Matrices for one run of the Experiments

With average accuracy of (83.33%) which is higher than the average accuracy achieved by JGAAP using the Juola Cross Entropy Classifier of (72.05%), the new feature used for classification proved successful. Keep in mind that experiment conducted using JGAAP was combination of different features and different classifiers, while experiments conducted on the BLN-Gram-TF-ITF were done using the new feature by itself and using the KNN classifier.

3.4 Authorship Identification using BLN-Gram-TF-ITF

3.4.1 Experimenting on Chat bot Corpus

A new version of the BLN-Gram-TF-ITF was created for this experiment; the new version did eliminate the spaces from the N-Gram created. The chat bot's corpus was set for six chat bots, 10 text files for each chat bot of 4KB each. Following the flow chart in Figure 14, the first step is to load the text files and preprocess them. Next, we extract the BLN-G-Gram-TF-ITF feature for each chat bot. With (K=5)-

Fold cross validation and KNN classification with N=4 the experiments outperformed the results obtained by Ali et al. using JGAAP [52].

Table 12 displays the results of applying a t-test to analyze results for statistical significance for JGAAP vs. BLN-Gram-TF-ITF average accuracy on chat bot's corpus.

Table 12: t-test results for statistical significance between JGAAP average accuracy and BLN-Gram-TF-ITF average accuracy

95% CI for difference	(0.0385, 0.4275)		
T-Test of difference	0 (vs. not =)	T-Value	2.59
P-Value	0.022	DF	13

With an average accuracy of 83.33% and a maximum accuracy of 91.67%, the new feature shows a promising start for further experiments on human corpora. Figure 26 shows the comparison between authorship experiments using JGAAP [17] and the new feature on the same chat bot's corpus.

Figure 26: BLN-Gram-TF-ITF vs. JGAAP average accuracy and maximum accuracy for Chat bot's corpus

3.4.2 Experimenting on Human Corpus

The BLN-Gram-TF-ITF feature was used on Chat bot's data and results compared to experiments with same corpus using JGAAP [105]. Seven books from

Gutenberg project [99] for six different authors were selected for this experiment. The books were:

- Emma by Jane Austen.
- Paradise Lost by John Milton.
- The Man who was Thursday by G. K. Chesterton.
- The Wisdom of Father Brown by G. K. Chesterton.
- The Parent's Assistant by Maria Edgeworth.
- Moby Dick by Herman Melville.
- Hamlet by William Shakespeare.

For Chesterton, samples from his own two books were used as one author, so for the seven books we will have six authors to classify over.

3.4.2.1 Data Preparation

The seven books from Gutenberg project [99] were used in these experiments. Since Chesterton was the author of two of those books, Chesterton data were extracted from his books equally. Each book has been split into text files based on word count. File sizes started with 50 words per file up to 1000 words per file [105]. Ten files were randomly selected from each group for experimenting. A total of 60 files were included in the corpus was created.

3.4.2.2 Experiment Plan

For each sample size, 20 different runs using K-cross validation with $K=5$ were conducted. A total of 100 outcomes per sample size was collected and compared.

The plan for experimenting was in three directions:

- a. Testing the BLN-Gram-TF-ITF on human English text, since it had been only implemented on chat bots so far.
- b. Changing file size based on word count.
- c. Changing the N value for N-Gram. Three values were selected for $N=3$, 4, and 5.

Figure 27 shows the average accuracy obtained for implementing the BLN-Gram-TF-ITF using trigrams for the different file sizes.

Figure 27: File size in word counts vs. average accuracy achieved for N=3

As expected, accuracy for small files is less compared to larger files. The overall accuracy of the proposed feature is notably high taking into consideration that the experiments only used this feature by itself.

Figure 28: File size in word counts vs. average accuracy achieved for N=4

Figure 28 represents the average accuracy for N=4, while Figure 29 represents the results for average accuracy for N=5.

Figure 29: File size in word counts vs. average accuracy achieved for N=5

To compare the three experiments when changing N, Figure 30 has all three line charts together. It is obvious that all charts follow the same pattern, lower accuracy for small files and as file size increases accuracy follows. For N=5 accuracy was higher for smaller files compared to using N=3 or 4, and was relatively the same as N=4 for files with 600 words. Overall, N=3 was the worst for small and large files and was similar to N=4 for files between 400-600 words.

Figure 30: File size in word counts vs. average accuracy achieved for N=3, 4, and 5

Figure 31 shows the box and whisker plot for the average accuracy over the three values of N.

Figure 31: Box and Whisker plot for the average accuracy for N=3, 4, and 5

Different file sizes affected the accuracy of identifying authors. Files with word counts exceeding 350 words showed accuracy of 84.5% over different values of N. At 200 words, the average accuracy achieved was 71.1% while at 100 words the average accuracy dropped to 63.3%.

From the experiments, accuracy was best when $N=5$ for small files and large files, while both $N=3$ and $N=4$ outperformed $N=5$ for files with word count between 400 – 600 words. Overall average accuracy for $N=3$ was 79.9% for all variable file size, 88.5% for $N=4$, and 85.3% for $N=5$ [105].

3.4.3 BLN-Gram-TF-ITF for Paragraph similarity

Measuring similarity between texts has been used for many applications. Plagiarism detection is one of the foremost applications in this regard. Cosine similarity is one of the most used algorithms for similarity measure as seen in Equation 9 [72].

Six Authors were selected for the experiments, and 10 paragraphs per author were used with word count of 500 words per paragraph. The corpus used the following books:

1. Bleak House by Charles Dickens
2. Mansfield Park by Jane Austen.
3. The Adventures of Tom Sawyer by Mark Twain
4. The Parent's Assistant by Maria Edgeworth.
5. Moby Dick by Herman Melville.
6. Hamlet by William Shakespeare.

One sample paragraph with 500 words was tested from each of the books and the results are shown in Table 13.

Table 13: Similarities between paragraphs from six different books

Book #	1	2	3	4	5	6
1	1	0.073	0.1192	0.1708	0.1877	0.1025
2	0.073	1	0.0657	0.2487	0.0737	0.0321
3	0.1192	0.0657	1	0.0872	0.1136	0.0484
4	0.1708	0.2487	0.0872	1	0.1082	0.0727
5	0.1877	0.0737	0.1136	0.1082	1	0.5856
6	0.1025	0.0321	0.0484	0.0727	0.5856	1

The results show a similarity between the paragraphs chosen from Melville and Shakespeare, which was confirmed by a work published for David Cope in 1999 [106]. Similarities between all other paragraphs are low [107].

3.4.4 Experiments with Combining BOW with BLN-Gram-TF-ITF

For this part of the experiment, the Bag Of Words (BOW) feature was combined with the BLN-Gram-TF-ITF to be implemented for this part of the experiments. Figure 32 describes the combining procedure for the BLN-Gram-TF-ITF with BOW. As seen from the flow chart, the BOW was assigned to the end of the BLN-Gram-TF-ITF.

Figure 33 shows that average accuracy did change dramatically for small files, and was fluctuating around the values for larger files compared to experiments using the BLN-Gram-TF-ITF alone [107].

Performing paired t-test to test for statistical significance, Table 14 shows that the values obtained after combining the BOW feature with the BLN-Gram-TF-ITF is actually statistically significant with P-Value of 0.002 [107].

Figure 32: The Byte Level N-Gram Term Frequency Inverse Token Frequency combined with BOW Flow chart

Table 14: Paired t-test for statistical significant between the two average accuracies of BLN-Gram-TR-ITF and the combined feature with BOW. 95% CI for mean difference :(-0.1314, -0.0362)

	N	Mean	StDev	SE Mean
C1	21	0.7469	0.1241	0.0271
C2	21	0.8306	0.0456	0.0099
Difference	21	-0.0838	0.1046	0.0228

t-test of mean difference = 0 (vs not = 0):

T-Value = -3.67 P-Value = 0.002

There were two changes to the feature creation for this part of the experiments, Stem Porter was used and the function words were removed. As discussed earlier, function words or stop words are good authorial feature but are not helpful when working with similarity and classification.

Figure 33: Average accuracy vs. file size between BLN-Gram-TF-ITF and combined with BOW

3.4.5 BLN-Gram-TF-ITF as Language Independent Feature

Ten different German language books were used, 10 files with 500 words each per book. The following are the books used in this experiment:

- Faust, Eine Tragödie by Johann Wolfgang von Goethe.
- Phantasten by Erich von Mendelssohn Release.
- Durch Wüste und Harem Gesammelte Reiseromane, Band I by Karl May.
- Der Untertan by Heinrich Mann.
- Buddenbrooks Verfall einer Familie by Thomas Mann.
- Das rasende Leben Zwei Novellen by Kasimir Edschmid.
- Die sechs Mündungen Novellen by Kasimir Edschmid.
- Die Fürstin by Kasimir Edschmid.
- Timur Novellen by Kasimir Edschmid.
- Über den Expressionismus in der Literatur und die neue Dichtung by Kasimir Edschmid.

This experiment was performed to investigate if BLN-Gram-TF-ITF is language independent. It was demonstrated to work on English language by Ali et al. [105].

The experiments on the German corpus yielded an average accuracy of 85%. This demonstrates that the BLN-Gram-TF-ITF is indeed language independent [107].

3.5 Bitcoins Mysterious Founder

In 2008, Bitcoin was introduced in a paper as a peer-to-peer payment system and a digital currency that was later very tempting for many of the e-commerce merchants due to the high security and very low transaction fees compared to credit cards [108, 109].

For now, the founder of Bitcoins is known as Satoshi Nakamoto [108], but there are no records for a person named Satoshi Nakamoto, and numerous speculations have been proposed for the true identity of the mysterious founder and creator of Bitcoins. Five different scientists has been proposed as possible founder(s) for the Bitcoins as listed below [110]:

- Michael Clear.
- Neal King.
- Shinichi Mochizuki
- Vladimir Oksman.
- Charles Bry.

Michael Clear, Neal King, and Charles Bry used a term found in Satoshi's paper for Bitcoins in a patent they jointly filed, "computationally impractical to reverse". So text data was collected for all candidates and experiments targeted toward similarities between any of those text authors with Satoshi's writing. For each candidate, three chunks of text files, 500 words each, were used to compare with others. The result was not conclusive enough to decide.

Table 15: Similarities between candidate authors

	Michael Clear	Neal King	Three Authors	Shinichi Mochizuki	Vladimir Oksman	Satoshi Nakamoto
Michael Clear	1.0 0.55 0.52 0.68 1.0					
Neal King	0.42 0.40 0.41 0.50 0.51 0.50 0.42 0.44 0.46	1.0 0.70 1.0 0.53 0.60 1.0				
Three Authors	0.30 0.31 0.31 0.33 0.32 0.32 0.30 0.32 0.32	0.33 0.33 0.30 0.34 0.35 0.31 0.33 0.35 0.32	1.0 0.88 1.0 0.74 0.78 1.0			
Shinichi Mochizuki	0.38 0.34 0.36 0.46 0.41 0.45 0.46 0.40 0.44	0.32 0.38 0.33 0.34 0.42 0.37 0.37 0.44 0.41	0.26 0.30 0.27 0.33 0.35 0.33 0.33 0.36 0.34	1.0 0.66 1.0 0.67 0.81 1.0		
Vladimir Oksman	0.43 0.46 0.45 0.43 0.42 0.43 0.38 0.41 0.43	0.50 0.53 0.50 0.50 0.55 0.54 0.49 0.53 0.51	0.38 0.39 0.38 0.34 0.33 0.34 0.32 0.31 0.34	0.36 0.41 0.41 0.35 0.41 0.42 0.32 0.37 0.39	1.0 0.76 1.0 0.60 0.69 1.0	
Satoshi Nakamoto	0.50 0.49 0.49 0.47 0.44 0.47 0.39 0.36 0.38	0.51 0.54 0.58 0.49 0.53 0.52 0.43 0.46 0.44	0.36 0.36 0.36 0.32 0.34 0.42 0.40 0.41 0.39	0.38 0.44 0.45 0.38 0.42 0.48 0.33 0.37 0.38	0.51 0.51 0.5 0.45 0.45 0.48 0.41 0.42 0.47	1.0 0.7 1.0 0.6 0.66 1

BLN-Gram-TF-ITF was used with cosine similarity to measure “who will be the best candidate for the mysterious Satoshi Nakamoto“?

Table 15 shows the cosine similarities between candidate authors. Three files per author were used in the experiments. Values greater than 0.5 have been shaded to reflect higher similarity. One can notice that King has the highest number of similar files with Satoshi next was Oksman. The similarity values are still not conclusive enough. Interestingly, King and Oksman have a high similarity between each other’s files.

The patent filed by Clear, King and Bry was used in the experiments as a training data for an additional candidate author. One of the possibilities was that all three candidates jointly could be the mysterious Satoshi. The results in Table 15 did not support this theory since the similarity between the three authors with Satoshi was low.

3.6 Quality of Writing (QoW) Features

Initially, the main goal was to study one’s writing samples and look for special features that could correlate his/her quality of writing with his/her Intelligence Quotient (IQ). Unfortunately, IQ scores were hard to collect, other than limited number of celebrities’ IQs available online. None of the contacted IQ groups, like Mensa [111], High IQ Society [112], The IQ Test[113] and the Cognitive IQ Society [114] agreed to release any of their members’ data, following the United States Privacy Act [115]. So unless this data is available, it will be impossible to continue this part of the research the way it was originally planned. So a simplification to the problem was proposed to look at the problem as a binary problem. The new strategy will try to classify smart versus below average person based on Quality of Writing.

To do this, a sample dataset was used from students’ writings for elementary, middle, and high schools [116, 117]. In addition, sample texts from Marilyn vos Savant’s quotes were used in this experiment. Marilyn vos Savant, an American magazine columnist and author who has a Guinness record for the highest IQ score (218) [118, 119]. And a set of randomly selected scientific published papers as another possible class, each text file was 400 words long.

3.6.1 Features used for Quality of Writing (QoW)

Many features could be used to reflect (QoW), the features selected in this research were the following:

1. **Hapax-Legomena:** Words that have a frequency of one.
2. **Dis-Legomena:** Words that have a frequency of two.
3. **Word's length:** Frequency of words based on length from 1 – 20 letters.
4. **Words ending with a vowel:** Frequency of words ending with a vowel 'aeoiu'.
5. **Words starting with a vowel:** Frequency of words starting with a vowel 'aeoiu'.
6. **Vocabulary richness:** Will measure the number of unique words divided by total number of words in the text file.

For each file in the training set, a feature for that file was saved and used to train the model using K-Nearest Neighbor classifier with K=3. Test files were stored in a separate folder. Another script was called and features extracted from these test files. Using the previously saved model, the predicted classes were displayed for each file in the test folder.

3.6.2 Scientific Writing Samples versus School Students Writing

The first set of experiments was to test 92 files, 46 files for scientific papers and 46 files for student samples. Each text file was 400 words. 80% of the files in each class were used for training. The files were stored in a separate folder. The remaining 20% of the files were used for testing (9 files for each class) and were stored in another folder. The average accuracy achieved was 98.9% using KNN with K=3.

3.6.3 Classifying Three Classes

To make things a little more challenging, the school students' writing was split based on grade and compared with the novel text as follows:

- a) Middle school students' writings.
- b) High School students' writings.
- c) Marilyn vos Savant (Very high IQ score author).

The total number of files used in this experiment was 42 files, with 14 files for each class. Splitting point was 80% training and 20% testing. Four files from each class were tested for a total of 12. The average accuracy achieved in this part was 66.67%.

3.6.4 Classifying Six Classes

Going further, more splitting to the school students' writing was made based on actual grades. Due to the lack of data, grades data were merged and the Kindergarten text was not used in the training. The classes were as followed:

- a) 2nd - 3rd grade text.
- b) 4th – 5th grade text.
- c) Middle school text.
- d) High school text.
- e) Marilyn vos Savant (Very high IQ score author).
- f) Scientific text.

Table 16: Six classes' prediction using (QoW) features

#	Test File	Predicted Class	Correct Class
1	4 th -5 th grade	4 th -5 th grade	Correct
2	4 th -5 th grade	4 th -5 th grade	Correct
3	4 th -5 th grade	4 th -5 th grade	Correct
4	4 th -5 th grade	4 th -5 th grade	Correct
5	High School	High School	Correct
6	High School	Middle school	False/ But Close
7	High School	High School	Correct
8	High School	Marilyn-Geniuse	False
9	Marilyn vos Savant	Marilyn-Geniuse	Correct
10	Marilyn vos Savant	Marilyn-Geniuse	Correct
11	Marilyn vos Savant	Marilyn-Geniuse	Correct
12	Marilyn vos Savant	Marilyn-Geniuse	Correct
13	Middle school	Middle school	Correct
14	Middle school	4 th -5 th grade	False/ But Close
15	Middle school	Marilyn-Geniuse	False
16	Middle school	4 th -5 th grade	False/ But Close
17	Scientific Text	Scientific Text	Correct
18	Scientific Text	Scientific Text	Correct
19	Scientific Text	Marilyn-Geniuse	False/ But Close
20	Scientific Text	Scientific Text	Correct
21	2 nd -3 rd grade	2 nd -3 rd grade	Correct
22	2 nd -3 rd grade	2 nd -3 rd grade	Correct
23	2 nd -3 rd grade	2 nd -3 rd grade	Correct

24	2 nd -3 rd grade	2 nd -3 rd grade	Correct
25	My Middle school daughter text	High School	False
26	My 3 rd grade son text	4 th – 5 th grade	False/ But Close
27	My supervisor text	Scientific Text	Correct
Actual Achieved Accuracy			70.37%
Realistic Accuracy			88.88%

A total of 60 text files were used for the training, with a size of 400 words each. Three extra files from my 3rd grade son, my 6th grade daughter, and my supervisor published text samples were tested. The results were as shown in Table 16.

The selected classes have a thin borderline for separation. Taking a closer look, a 5th grade student is almost a middle school student. Also, a student in the 3rd grade is expected to be very close to the 4th grade student. This will make classes hard to isolate, and accuracy is expected to be low in this kind of data. Nevertheless, the accuracy achieved was 70.37%, but if one investigated the results, counting-in the results marked as close, i.e. when predicting middle school text as 4th -5th grade, or even my 3rd grade son who was predicted as 4th-5th grade, if we count all that as correct, then we will get a total accuracy of 88.88% which for this kind of data is more realistic.

CHAPTER IV

CONCLUSION AND FUTURE WORK

5.1 Authorship Identification and the BLN-Gram-TF-ITF

Authorship attribution is one of the most appealing and attractive fields of research. Big room for improvements is still available due to the lack of confidence in current obtained accuracy. Authorship methodologies did not fulfill the requirements for applications to be publically used in the courts of law or wherever a need is found. Thousands of features have been proposed for authorship attribution, various studies on implementations, text size and number of candidate authors.

A new feature has been proposed and tested on chat bots' text, human English text and German books. It has been also implemented as a feature for similarity between paragraphs and between books. The Byte Level N-Gram Term Frequency Inverse Token Frequency was demonstrated as a successful feature for all the above implementations. Combining this feature with another feature has been implemented and accuracy has been investigated. A combination with the Bag Of Words (BOW) feature was implemented and a noticeable improvement for short text files was demonstrated.

BLN-Gram-TF-ITF was validated as language independent when experimenting on German books and obtained an average accuracy of 85% identifying the correct author from a list of 10 candidate authors.

The mystery of Bitcoins' founder remains unrevealed. The experiments did show that Neal King has the highest similarity to Satoshi Nakamoto the mysterious name behind Bitcoins. But since truth is not revealed up to this moment, no one can claim

success of finding the right Satoshi. In future if Neal King claim himself as Satoshi then experiments conducted in this research will be correct.

5.2 Quality of Writing versus IQ

Correlating Quality of Writing (QoW) to ones IQ is a very interesting aspect of this research; actually it is the most interesting one of them all. It was very hard to collect IQ scores since there is no publicly available dataset. A simplification was made to the plan, by looking at the problem as a binary problem, then stepping up.

Classifying text based on quality of writing using six different features was a great success for identifying scientific versus school students' writings. An average of 98.9% success was achieved for a dataset of 92 text files with the size of 400 words each.

Increasing the number of classes to three was a bit challenging; it is hard to correctly claim that a 5th grade student's writing is very different from that of a 6th grade student, while they belong to two different classes in this experiment. The three classes were: middle school, high school and a text from a novel. The accuracy achieved in this part was 66.67%.

The more the number of classes, the more challenging the problem gets. Six classes were selected; the classes were: 2nd -3rd grade, 4th -5th grade, middle school, high school, sample texts from a High IQ record holder, and scientific text. As seen in Table 16, the actual accuracy was 70.33%, but if one investigates the close grades and consider those correct we will get an accuracy of 88.88%.

More work is needed to investigate combining BLN-Gram-TF-ITF with more features using different classifiers. The results of the Quality of Writing experiments were interesting. Deeper investigation is required to experiment with additional features and different combinations. More looking into the right combination to correlate one's writing to the IQ level of that same person. GRE score could be a valid alternative for the IQ score, since this score is easier to obtain.

More work on similarity measures is needed to make the separation between classes clearer and more accurate.

REFERENCES

- [1] A. Jain, "Biometric Identification," *Communications of the ACM*, vol. 43, pp. 91-98, 2000.
- [2] A. Jain, A. A. Ross, and K. Nandakumar, *Introduction to Biometrics*: Springer-Verlag New York, LLC, 2011.
- [3] A. Ross and A. K. Jain, "Multimodal Biometrics: An Interview," in *12th European Signal Processing Conference (EUSIPCO)*, Vienna, Austria, 2004, pp. 1221-1224.
- [4] R. V. Yampolskiy and V. Govindaraju, "Behavioral Biometrics: a Survey and Classification," *International Journal of Biometrics (IJBM)*. vol. 1, pp. 81-113, 2008.
- [5] O. Angela, "An Instant Messaging Intrusion Detection System Framework: Using character frequency analysis for authorship identification and validation," in *40th Annual IEEE International Carnahan Conferences Security Technology*, Lexington, KY, 2006, pp. 160-172.
- [6] P. Juola, "Authorship attribution," *Foundations and Trends® in Information Retrieval*, vol. 1, pp. 233-334, 2007.
- [7] T. C. Mendenhall, "THE CHARACTERISTIC CURVES OF COMPOSITION," *Science*, vol. ns-9, pp. 237-246, March 11, 1887 1887.
- [8] G. K. Zipf, *Selected studies of the principle of relative frequency in language*. Cambridge, Mass.: Harvard University Press, 1932.
- [9] G. U. Yule, "On Sentence-Length as a Statistical Characteristic of Style in Prose: With Application to Two Cases of Disputed Authorship," *Biometrika*, vol. 30, pp. 363-390, 1939.
- [10] G. U. Yule, "The Statistical study of literary vocabulary," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 42, pp. b1-b2, 1944.
- [11] R. Arun, V. Suresh, and C. E. V. Madhavan, "Stopword Graphs and Authorship Attribution in Text Corpora," pp. 192-196, 2009.
- [12] E. Stamatatos, "A survey of modern authorship attribution methods," *Journal of the American Society for Information Science and Technology*, vol. 3, pp. 538-556, 2008.
- [13] Y. Wu, G. Wang, W. Li, and Z. Li, "Automatic Chatbot Knowledge Acquisition from Online Forum via Rough Set and Ensemble Learning," pp. 242-246, 2008.
- [14] S. G. Efimovich and S. O. Gennadyevich, "Automatic search of indicators of text authorship," in *The 7th Korea-Russia International Symposium on Science and Technology.*, KORUS, 2003, pp. 185-188.
- [15] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, pp. 1-47, 2002.
- [16] D. I. HOLMES, "The evolution of stylometry in humanities scholarship," *Literary and Linguistic Computing*, vol. 13, pp. 111-117, 1998.
- [17] P. Juola. (2011, July, 4th). *Java Authorship Attribution Application*. Available: http://evllabs.com/jgaap/w/index.php/Main_Page

- [18] M. Koppel, S. Argamon, and A. Shimoni, "Automatically Categorizing Written Texts by Author Gender," *Literary and Linguistic Computing*, vol. 17, p. in press., 2003.
- [19] S. Argamon and S. Levitan, "Measuring the usefulness of function words for authorship attribution," in *Joint Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing*, 2005.
- [20] J. F. Burrows, "Word patterns and story shapes: The statistical analysis of narrative style.," *Literary and Linguistic Computing*, vol. 2, pp. 61-70, 1987.
- [21] A. Abbasi and H. Chen, "Applying authorship analysis to extremist-group Web forum messages," *IEEE Intelligent Systems*, vol. 20(5), pp. 67-75, 2005.
- [22] S. Argamon, M. Saric, and S. Stein, "Style mining of electronic messages for multiple authorship discrimination: First results.," *9th ACM SIGKDD*, pp. 475-480, 2003.
- [23] Z. Y. and J. Zobel, "Effective and scalable authorship attribution using function words.," *2nd Asia Information Retrieval Symposium.*, 2005.
- [24] M. Koppel and J. Schler, "Exploiting stylistic idiosyncrasies for authorship attribution.," *IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis*, pp. 69-72, 2003.
- [25] S. Argamon, C. Whitelaw, P. Chase, S. R. Hota, N. Garg, and S. Levitan, "Stylistic text classification using functional lexical features," *Journal of the American Society for Information Science and Technology*, vol. 58, pp. 802-822, 2007.
- [26] F. Peng, D. Shuurmans, and S. Wang, "Augmenting naive Bayes classifiers with statistical language models.," *Information Retrieval Journal*, vol. 7, pp. 317-345, 2004.
- [27] Sanderson and G. C., S., "Short text authorship attribution via sequence kernels, Markov chains and author unmasking: An investigation," *International Conference on Empirical Methods in Natural Language Engineering*, pp. 482-491, 2006.
- [28] R. M. Coyotl-Morales, L. Villaseñor-Pineda, M. Montes-y-Gómez, and P. Rosso, "Authorship attribution using word sequences," *11th Iberoamerican Congress on Pattern Recognition*, pp. 844-853, 2006.
- [29] J. da Silva and G. Lopes, "A local maxima method and a fair dispersion normalization for extracting multi-word units from corpora," in *Sixth Meeting on Mathematics of Language*, 1999.
- [30] J. Houvardas and E. Stamatatos, "N-Gram Feature Selection for Authorship Identification," in *Artificial Intelligence: Methodology, Systems, and Applications*. vol. 4183, J. Euzenat and J. Domingue, Eds., ed: Springer Berlin Heidelberg, 2006, pp. 77-86.
- [31] M. Koppel, J. Schler, and S. Argamon, "Computational methods in authorship attribution," *Journal of the American Society for Information Science and Technology*, vol. 60, pp. 9-26, 2008.
- [32] J. Fürnkranz, "A study using n-gram features for text categorization," *Austrian Research Institute for Artificial Intelligence*, vol. 3, pp. 1-10, 1998.

- [33] A. Mohan, I. M. Baggili, and M. K. Rogers, "Authorship attribution of SMS messages using an N-grams approach," CERIAS Tech Report 20112010.
- [34] O. De Vel, A. Anderson, M. Corney, and G. Mohay, "Mining e-mail content for author identification forensics," *ACM Sigmod Record*, vol. 30, pp. 55-64, 2001.
- [35] R. Zheng, J. Li, H. Chen, and Z. Huang, "A framework for authorship identification of online messages: Writing - style features and classification techniques," *Journal of the American Society for Information Science and Technology*, vol. 57, pp. 378-393, 2005.
- [36] T. Matsuura and Y. Kanada, "Extraction of authors' characteristics from Japanese modern sentences via n-gram distribution," in *Discovery Science*, 2000, pp. 315-319.
- [37] E. Stamatatos, "Author Identification Using Imbalanced and Limited Training Texts," pp. 237-241, 2007.
- [38] H. Baayen, H. Van Halteren, and F. Tweedie, "Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution," *Literary and Linguistic Computing*, vol. 11, pp. 121-132, 1996.
- [39] E. Loper and S. Bird, "NLTK: The natural language toolkit," in *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, 2002, pp. 63-70.
- [40] S. Argamon-Engelson, M. Koppel, and G. Avneri, "Style-based text categorization: What newspaper am I reading," in *Proc. of the AAAI Workshop on Text Categorization*, 1998, pp. 1-4.
- [41] O. Kukushkina, A. Polikarpov, and D. V. Khmelev, "Using literal and grammatical statistics for authorship attribution," *Problems of Information Transmission*, vol. 37, pp. 172-184, 2001.
- [42] J. Diederich, J. Kindermann, E. Leopold, and G. Paass, "Authorship attribution with support vector machines," *Applied intelligence*, vol. 19, pp. 109-123, 2003.
- [43] M. Gamon, "Linguistic correlates of style: authorship classification with deep linguistic analysis features," in *Proceedings of the 20th international conference on Computational Linguistics*, 2004, p. 611.
- [44] Y. Zhao and J. Zobel, "Searching with style: Authorship attribution in classic literature," in *Proceedings of the thirtieth Australasian conference on Computer science-Volume 62*, 2007, pp. 59-68.
- [45] J. Leskovec and J. Shawe-Taylor, "Semantic text features from small world graphs," 2005.
- [46] M. A. K. Halliday, "Functional grammar," *London: Edward Arnold*, 1994.
- [47] M. Koppel, J. Schler, S. Argamon, and E. Messeri, "Authorship attribution with thousands of candidate authors," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 659-660.
- [48] Chatbot. (2011, June, 2011). *Chatbot - Artificial person with interactive textual conversation skills*. Available: www.chatbot.org/chatbot/

- [49] Webopedia. (2011, June 20). *What is chat bot? A Word Definition from the Webopedia Computer Dictionary*. Available: www.webopedia.com/TERM/C/chat_bot.html
- [50] Wikipedia. (2011, June 22). *Chatterbot- Wikipedia, the free encyclopedia*. Available: www.en.wikipedia.org/wiki/Chatterbot
- [51] N. Ali, M. Hindi, and R. V. Yampolskiy, "Evaluation of authorship attribution software on a Chat bot corpus," in *Information, Communication and Automation Technologies (ICAT), 2011 XXIII International Symposium on*, 2011, pp. 1-6.
- [52] N. Ali, M. Hindi, and R. V. Yampolskiy, "Evaluation of authorship attribution software on a Chat bot corpus," in *XXIII International Symposium on Information, Communication and Automation Technologies (ICAT)*, Sarajevo, Bosnia and Herzegovina, 2011, pp. 1-6.
- [53] N. Ali, D. Schaeffer, and R. V. Yampolskiy, "Linguistic Profiling and Behavioral Drift in Chat Bots," in *Midwest Artificial Intelligence and Cognitive Science Conference*, 2012, p. 27.
- [54] B. A. Shawar and E. Atwell, "A comparison between ALICE and Elizabeth chatbot systems," Technical report, School of Computing, University of Leeds 2002.
- [55] AI-Research. (2012, Dec, 8). *Creating a new form of life- Chatbots Explained*. Available: http://www.a-i.com/show_tree.asp?id=61&level=3&root=1
- [56] J. Weizenbaum, "ELIZA---a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, pp. 36-45, 1966.
- [57] ALICE. (2011, June 12). *ALICE*. Available: <http://alicebot.blogspot.com/>
- [58] S. Gianvecchio, M. Xie, Z. Wu, and H. Wang, "Humans and Bots in Internet Chat: Measurement, Analysis, and Automated Classification," *Networking, IEEE/ACM Transactions on*, vol. PP, pp. 1-1, 2011.
- [59] P. J. Y. Jonathan, C. C. Fung, and K. W. Wong, "Devious Chatbots - Interactive Malware with a Plot Progress in Robotics." vol. 44, J.-H. Kim, S. S. Ge, P. Vadakkepat, N. Jesse, A. Al Manum, S. Puthusserypady K, U. Rückert, J. Sitte, U. Witkowski, R. Nakatsu, T. Braunl, J. Baltes, J. Anderson, C.-C. Wong, I. Verner, and D. Ahlgren, Eds., ed: Springer Berlin Heidelberg, 2009, pp. 110-118.
- [60] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Who is tweeting on Twitter: human, bot, or cyborg?," presented at the Proceedings of the 26th Annual Computer Security Applications Conference, Austin, Texas, 2010.
- [61] C. Sporleder and A. Lascarides, "Combining hierarchical clustering and machine learning to predict high-level discourse structure," in *Proceedings of the 20th international conference on Computational Linguistics*, 2004, p. 43.
- [62] W. C. Mann and S. A. Thompson, "Rhetorical structure theory: Toward a functional theory of text organization," *Text*, vol. 8, pp. 243-281, 1988.
- [63] R. Soricut and D. Marcu, "Sentence level discourse parsing using syntactic and lexical information," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, 2003, pp. 149-156.

- [64] D. Marcu, *The theory and practice of discourse parsing and summarization*: MIT Press, 2000.
- [65] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, pp. 513-523, 1988.
- [66] W. Lam and C. Y. Ho, "Using a generalized instance set for automatic text categorization," in *Annual ACM Conference on Research and Development in Information Retrieval: Proceedings of the 21 st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998, pp. 81-89.
- [67] A. Bergo, "Text categorization and prototypes," 2002.
- [68] T. Joachims, "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization," DTIC Document 1996.
- [69] E. Stamatatos, "Intrinsic plagiarism detection using character n-gram profiles," *threshold*, vol. 2, pp. 1,500, 2009.
- [70] M. Koppel, N. Akiva, E. Alshech, and K. Bar, "Automatically classifying documents by ideological and organizational affiliation," 2009.
- [71] M. Koppel and J. Schler, "Authorship verification as a one-class classification problem," presented at the Proceedings of the twenty-first international conference on Machine learning, Banff, Alberta, Canada, 2004.
- [72] Wikipedia. (2014, March). *Cosine similarity*. Available: http://en.wikipedia.org/wiki/Cosine_similarity
- [73] Wikipedia. (2012, Dec, 8th). *Turing Test*. Available: http://en.wikipedia.org/wiki/Turing_test
- [74] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, pp. 433-460, 1950.
- [75] H. G. Loebner. (2012, Jan 3). *Home Page of The Loebner Prize*. Available: <http://loebner.net/Prize/loebner-prize.html>
- [76] Wikipedia. (2014). *Frye Standard*. Available: http://en.wikipedia.org/wiki/Frye_standard
- [77] Wikipedia. (2014). *Daubert Standard*. Available: http://en.wikipedia.org/wiki/Daubert_standard
- [78] P. Juola and J. Sofko, "Proving and Improving Authorship Attribution Technologies," in *Proceedings of Canadian Symposium for Text Analysis (CaSTA)*, 2004.
- [79] M. Hindi, "Chat Bot Reloaded," vol. 1, ed, 2011.
- [80] Pace_University. (2011, June, 4th). *Stylometry*. Available: <http://utopia.csis.pace.edu/cs615/2006-2007/team2/>
- [81] CleverBot. (2011, July, 5th). *CleverBot* Available: <http://cleverbot.com/>
- [82] HAL. (2011, June 16). *AI Research*. Available: http://www.a-i.com/show_tree.asp?id=97&level=2&root=115
- [83] Jeeney. (2011, March, 11). *Artificial Intelligence Online*. Available: <http://www.jeeney.com/>
- [84] SkyNet. (2011, April, 20). *SkyNet - AI*. Available: <http://home.comcast.net/~chatterbot/bots/AI/Skynet/>
- [85] TalkBot. (2011, April 14). *TalkBot- A simple talk bot*. Available: <http://www.frontiernet.net/~wcowart/chatbot.html>

- [86] C. TalkBot. (2011). *TalkBot- A simple talk bot*. Available: <http://www.chatbots.org/chatbot/talk-bot/>
- [87] M. B. Malyutov, "Authorship attribution of texts: a review," *Electronic Notes in Discrete Mathematics*, vol. 21, pp. 353-357, 2005.
- [88] C. P. Smith, *Motivation and personality: Handbook of thematic content analysis*: Cambridge University Press, 1992.
- [89] D. McAdams, *Explorations in personality*: Oxford University Press, USA, 2007.
- [90] D. G. Winter, "Measuring the motives of political actors at a distance," *The psychological assessment of political leaders*, pp. 153-177, 2003.
- [91] D. K. Simonton, "The "other IQ": Historiometric assessments of intelligence and related constructs," *Review of General Psychology*, vol. 13, p. 315, 2009.
- [92] M. Kantardzic, *Data mining: concepts, models, methods, and algorithms*: Wiley-IEEE Press, 2011.
- [93] S. Gianvecchio, X. Mengjun, W. Zhenyu, and W. Haining, "Humans and Bots in Internet Chat: Measurement, Analysis, and Automated Classification," *Networking, IEEE/ACM Transactions on*, vol. 19, pp. 1557-1571, 2011.
- [94] J. Ramos, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the First Instructional Conference on Machine Learning*, 2003.
- [95] V. Kešelj, F. Peng, N. Cercone, and C. Thomas, "N-gram-based author profiles for authorship attribution," in *Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING*, 2003, pp. 255-264.
- [96] G. Frantzeskou, E. Stamatatos, S. Gritzalis, C. E. Chaski, and B. S. Howald, "Identifying authorship by byte-level n-grams: The source code author profile (scap) method," *International Journal of Digital Evidence*, vol. 6, pp. 1-18, 2007.
- [97] L. Von Ahn, M. Blum, and J. Langford, "Telling humans and computers apart automatically," *Communications of the ACM*, vol. 47, pp. 56-60, 2004.
- [98] R. V. Yampolskiy and V. Govindaraju, "Embedded noninteractive continuous bot detection," *Computers in Entertainment (CIE)*, vol. 5, p. 7, 2008.
- [99] Gutenberg. (2012, Dec, 20). *Project Gutenberg*. Available: <http://www.gutenberg.org>
- [100] Jabberwacky. (2011, June, 10). *Jabberwacky-live chat bot-AI Artificial Intelligence chatbot*. Available: <http://www.jabberwacky.com/>
- [101] Jabberwock. (2011, June, 12). *Jabberwock Chat*. Available: <http://www.abenteuermedien.de/jabberwock/>
- [102] RapidMiner. (2011, Dec 20). *Rapid- I*. Available: <http://rapid-i.com/>
- [103] Python. (2012, Jan, 10). *Python Programming Language- Official Website*. Available: www.python.org
- [104] NLTK. (2012, Jan,10). *Natural Language Toolkit- NLTK 2.0 documentation*. Available: www.nltk.org
- [105] N. Ali and R. V. Yampolskiy, "BLN-Gram-TF-ITF as a new Feature for Authorship Identification," in *The Third ASE International Conference on Cyber Security*, Stanford, CA, USA, 2014.

- [106] (1999). *Melville / Shakespeare*. Available:
https://cms.grcc.edu/sites/default/files/docs/shakespeare/contemporary/melville_shakespeare.pdf
- [107] N. Ali and R. V. Yampolskiy, "BLN-Gram-TF-ITF as a Language Independent Feature for Authorship Identification and Paragraph Similarity," presented at the 9th Cyber and Information Science research Conference, Oak Ridge, Tennessee, 2014 (Under Review).
- [108] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Consulted*, vol. 1, p. 2012, 2008.
- [109] Wikipedia. (2014, August). *Bitcoin*. Available:
<http://en.wikipedia.org/wiki/Bitcoin>
- [110] C. Desk. (2013, August). *Who is Satoshi Nakamoto? The creator of Bitcoins remain elusive*. Available: <http://www.coindesk.com/information/who-is-satoshi-nakamoto/>
- [111] Mensa. (2013, March). *American Mensa, Ltd.* Available:
<http://www.us.mensa.org/>
- [112] (2012, December). *International High IQ Society*. Available:
<http://highiqsociety.org/>
- [113] (2013, December). *The IQ Test*. Available: <http://iqtest.com/>
- [114] (2013, December). *Cogn-IQ Online Intelligence testing*. Available:
<http://www.cogn-iq.org/>
- [115] U. D. o. Justice. (2014). *The Privacy Act of 1974*. Available:
<http://www.justice.gov/opcl/privstat.htm>
- [116] E. Week. (2014, February). *Writing samples*. Available:
http://www.edweek.org/media/25common_2c.pdf
- [117] T. T. M. Sense. (2014, January). *Writing Samples*. Available:
[http://www.ttms.org/PDFs/03%20Writing%20Samples%20v001%20\(Full\).pdf](http://www.ttms.org/PDFs/03%20Writing%20Samples%20v001%20(Full).pdf)
- [118] Wikipedia. (2014). *Marilyn vos Savant*. Available:
http://en.wikipedia.org/wiki/Marilyn_vos_Savant
- [119] IQ. (2014). *Estimated IQ for the Greatest Geniuses*. Available:
<http://hem.bredband.net/b153434/Index.htm>

Appendix A

This method will construct the Byte N Gram depending on N value

```
def byte_N_gram(N,text_file):  
    return [text_file[i:i+N] for i in range(len(text_file)-N+1)]
```

“”” What exactly does this method do is taking the text file or list of strings and the N value that we need to make as our sliding window for the N-Gram and loops over the list and generates the list with N length terms.”””

Appendix B

Returns the frequency of a term from a list of words

```
def freq(term, listOfWorks):
```

```
    return ' '.join(listOfWorks).count(term)
```

```
    """ This method will look for the occurrences of the term in list of words. """
```

Returns the number of tokens in a document or a text string

```
def word_count(doc):
```

```
    return len(doc)
```

```
    """ This method will return the length of the document, meaning how many
tokens that document have. """
```

Returns the relative term frequency to the total number of Tokens in a document

```
def tf(term, listOfWorks):
```

```
    return (freq(term, listOfWorks) / float(word_count(listOfWorks)))
```

```
    """ This method will call the freq method and the word_count method and divide
the outcome from the first method over the outcome of the second method """
```

#Returns the Number of tokens containing the term

```
def num_tokens_containing(term, list_of_tokens):
```

```
    count=0
```

```
    if ' '.join(list_of_tokens).count(term)>0:
```

```
        count= ' '.join(list_of_tokens).count(term)
```

```
    else:
```

```
        count= 1+ ' '.join(list_of_tokens).count(term)
```

```
    return count
```

```
    """ This method will count the number of times the term will occur in the list of
tokens, if equals to zero will return 1, otherwise will return the actual count. """
```



```

# Returns the Inverse Document Frequency score
def itf(term, list_of_Tokens):
    return math.log(len(list_of_Tokens) /
                    float(num_tokens_containing(term, list_of_Tokens)))
""" This method will return the Inverse Token Frequency (ITF)

# Calculates the TF-ITF
def tf_itf(term, N_gram_list, list_of_Tokens):
    return (tf(term, N_gram_list) * itf(term, list_of_Tokens))
""" This method will call the tf and the itf methods and multiply them together
and return the outcome"""

```

Appendix C

“”” The dynamic array declaration containing the feature set”””

```
tfidf_Feature_set=[[0 for x in xrange(len(finalCompletList[0]))]
                   for x in xrange(Corpus_length)]
```

“””The loop that will loop over each file in the corpus and call the tf_idf method and save the result in the tfidf_Feature_set array.”””

```
for i in range(Corpus_length):
    for term in finalCompletList[0]:
        tfidf_Feature_set[i][finalCompletList[0].index(term)]=tf_idf(term,
                               outCSV[i], outCSV2)
```

“”” This code snippet will do the saving to CSV file, and will create a file named “author1.csv, author2.csv, etc.”””

```
for i in range(numberOfAuthors):
    myfile=open('author'+str(i+1)+'.csv','wb')
    wr = csv.writer(myfile)
    for j in range(10):
        wr.writerow(tfidf_Feature_set[(i*10)+j])
    myfile.close()
```

Appendix D

“”” This is the Matlab code used for Classification and K-Fold cross validation, numAuthors is basically the Number of authors the corpus have. So the first loop will read the CSV files for each author. After that the values for K cross validation is set to 5, and the knn value is set to 4. The crossvalind function will return the indices that we are using when splitting the data into testing and training.”””

```
numAuthors=6;

for i=1:numAuthors

    filename=['csvoutput' num2str(i) '.csv'];

    Doc[4]=csvread(filename,0,0);

end

# The K fold value
K=5;
#The KNN with N = 4
knn=4;

for i=1:numAuthors

    N=size(Doc,1);

    indices{i}[1:19]= crossvalind('Kfold', N, K);

end

accuracy=[];
```

“”” This loop will set the test set and the training set and the labels (class name or author name) for each set”””

```
for j=1:K

    test=[];

    train=[];
```

```

testlabels=[];

trainlabels=[];

for i=1:numAuthors

    test=[test; Doc{i}(indices{i}==j,:)];

    testlabels=[testlabels i*ones(1,size(Doc{i}(indices{i}==j,:),1))];

    train=[train; Doc{i}(indices{i}~=j,:)];

    trainlabels=[trainlabels i*ones(1,size(Doc{i}(indices{i}~=j,:),1))];

end

```

“”” After setting the Test set and Train set with their labels we call the knnclassify to do the classification. And the confusionmat function will give us the confusion matrix of the experiments”””

```

predClass = knnclassify(test, train, trainlabels, knn,'cosine');

c{j} = confusionmat(testlabels,predClass)';

“”” Calculating the Accuracy”””

accuracy=[accuracy;sum(diag(c{j}))/sum(sum(c{j}))];

end

“”” Displaying the outputs to screen”””
disp('Max accuracy achieved is:')

disp(max(accuracy))

disp('Average accuracy achieved is:')

disp(mean(accuracy))

```

CURRICULUM VITAE

Nawaf T. Ali

2902 Patti Ln.

Louisville, KY. 40299

(502)-303-5185

nawaf.ali@louisville.edu

Objectives:

To obtain a challenging full time position to utilize computer engineering research skills in today's business and industrial market.

Projects:

- BLN-Gram TF-ITF, a new feature for Authorship Attribution, 2014.
- Analyzing Chat bots for Linguistic style and Behavioral Drift, 2011.
- Identifying Chat bots by their style, 2011.

Educational Qualifications:

- Ph.D. Candidate, 2013. GPA (3.36)

Cyber Security Research Lab.

Computer Engineering and Computer Science -J.B. Speed School of Engineering.

University of Louisville, KY, USA.

- Master in Computer Science, 2005. GPA (3.09)

Al-Balqa' Applied University (BAU), Jordan.

- B.Sc. in Information Technology (IT), 2001. GPA (3.59)

Al-Balqa' Applied University (BAU), Jordan.

- Diploma in Industrial Automation, 1993. (Very good – Not on a 4.0 scale)

Amman University College for Applied Engineering, Jordan.

Summary of Qualifications:

Skills:

- Work well under pressure.
- Communicate and co-ordinate tasks with others in a perfect manner.

- Take a considerable pride in the quality of my work, and always meet my deadlines.

▪ **Programming Languages:**

1. C/C++ programming.
2. Python.
3. Java.
4. Matlab.
5. HTML.
6. ECL.
7. Java Script /PHP
8. MySQL

▪ **Linguistic Proficiencies:**

1. Fluent in Arabic.
2. Fluent in English with TOEFL scores of 247 CBT, Oct 2002.

▪ **Published Papers:**

1. Evaluation of authorship attribution software on a Chat bot corpus.

N Ali, M Hindi, RV Yampolskiy - Information, Communication and Automation Technologies (ICAT), 2011 XXIII.

2. Linguistic Profiling and Behavioral Drift in Chat Bots.

N Ali, D Schaeffer, RV Yampolskiy - Midwest Artificial Intelligence and Cognitive Science Conference, MAICS-2012.

3. Behavioral Biometrics: Categorization and Review.

N Ali, D D'souza, A Mohamad, R Yampolskiy, IJNCR 2013 Journal.

4. BLN-Gram-TF-ITF as a New Feature for Authorship Identification.

N Ali, M Price, R Yampolskiy, CyberSecurity 2014. Stanford, CA, USA. (Under review)

▪ **Posters and Presentations:**

1. Engineering Expo, 2011. University of Louisville.

“Chat Bot Identification via Authorship Attribution”, (Poster).

2. Computer Engineering and Computer Science Seminar Series.

“BLN-Gram-TF-ITF for Chat bot Authorship Identification”.

3. Engineering Expo, 2013. University of Louisville.
“Identifying Chat bots using BLN-Gram-TF-ITF”, (Poster).
4. Graduate Research Symposium, 2013. University of Louisville.
“Identifying Chat bots using BLN-Gram-TF-ITF”, (Poster).

- **Certificates:**

- HPCC Introduction to THOR (June 2013).
- HPCC Introduction to ECL (June 2013).
- College Reading and Learning Association (CRLA) Level-II Advanced Certified Tutor (2012).
- College Reading and Learning Association (CRLA) Level-I Advanced Certified Tutor (2011).

- **Work Experiences:**

- August 2013 – Up to Date.

Senior Software Engineer.

Predictive Business Intelligence (PBI), Louisville, KY, USA

Job Description:

- Planning and organize the current and future development for PBI.
- Support and modify PBI applications when needed.
- Web master for the PBI website.

- August 2011 – June 2013.

Graduate Student Assistant and webmaster.

Resources for Academic Achievements (REACH), University of Louisville, KY.

Job Description:

Working as a Graduate Student Assistant (GSA) involves the following activities:

- ✓ Supervise tutors and make sure that they are doing their job in the right way.
- ✓ Work with other GSA's and make sure the intended goal is to be achieved.
- ✓ Make sure that all computers are ready for students at any time, and fix the ones that need to be fixed.

Working as webmaster for REACH website which involves:

- ✓ Maintain the REACH website (www.reach.louisville.edu).
 - ✓ Make updates whenever needed.
 - ✓ Make changes (Adding/Deleting) contents when ever asked for.
 - ✓ Come up with new designs for the website from time to time.
- March 2009 - November 2009.

University of Louisville, Web mining lab.

Job Description:

I worked on administrating and reconstructing websites for the lab projects.

Mainly were projects for NSF and research related to that project.

- August 2005 - July 2007.

Al-Balqa Applied University, Prince Abdullah Bin Ghazi Faculty of Science and Information Technology, IT Department, Jordan.

Job Description:

Teaching Assistant, giving different courses for the bachelor level in the IT and Computer Science Department.

- December 2001 – August 2005.

Al-Balqa Applied University, Prince Abdullah Bin Ghazi Faculty of Science and Information Technology, IT Department, Jordan.

Job Description:

Lab supervisor, taking care of all PC's in the lab by means of Software and Hardware, and helping the instructors in any lecture taken in the lab.

- October 1996 – October 2001.

Water Authority of Jordan (WAJ)-Zai Water Treatment Plant, Jordan.

Job Description:

Operator and Electrical Technician. My good knowledge in Computers enabled me to do some maintenance and doing the monthly report of the plant.

- October 1995 – October 1996.

National Industries Company (NIC), Jordan.

Job Description:

Electrical Technician.

- May 1994 – November 1995.

National Multi Industries Company (NAMECO), Jordan.

Job Description:

- ✓ Electrical Technician.
 - ✓ Operator in the Enameled Wires Factory.
 - ✓ Quality Control.
 - ✓ Shift Supervisor.
- September 1993 – April 1994.

Modern Mills and Macaroni Factories, Jordan.

Job Description:

Operator and Electrical Technician.

▪ **Training Courses:**

1. Introduction to SQL PL/SQL. December 2002-October 2002 for 90 hours.

Al-Balqa Applied University, Computer Center.

2. Oracle Reports. December 2002-October 2002 for 90 hours.

Al-Balqa Applied University, Computer Center.

3. Oracle Forms A and B. December 2002-October 2002 for 90 hours.

Al-Balqa Applied University, Computer Center.

▪ **Hobbies:**

- ✓ Research.
- ✓ Working with Design oriented software, and Websites.
- ✓ Calligraphy and Drawing
- ✓ Playing Basketball and Karate.