MANAGEMENT OF TECHNOLOGY

MASTER OF SCIENCE THESIS REPORT

An Exploratory Study on Authorship Verification Models for Forensic Purpose

Zhenshi Li

Date July, 2013

TITLE PAGE

TITLE: AN EXPLORATORY STUDY ON AUTHORSHIP VERIFICATION METHODS FOR FORENSIC PURPOSE

AUTHOR: ZHENSHI LI

STUDENT NUMBER: 4182340

DATE: JULY, 2013

EMAIL: Z.LI-2@STUDENT.TUDELFT.NL

FACULTY: TECHNOLOGY, POLICY AND MANAGEMENT

DEPARTMENT: INFORMATION AND COMMUNICATION TECHNOLOGY

GRADUATION COMMITTEE

CHAIRMAN: YAO-HUA TAN (INFORMATION AND COMMUNICATION TECHNOLOGY)

FIRST SUPERVISOR: JAN VAN DEN BERG (INFORMATION AND COMMUNICATION TECHNOLOGY)

SECOND SUPERVISOR: MAARTEN FRANSSEN (PHILOSOPHY)

EXTERNAL SUPERVISOR: COR J. VEENMAN (NETHERLANDS FORENSIC INSTITUTE)

GRADUATE INTERNSHIP ORGANIZATION: KECIDA (KNOWLEDGE AND EXPERTISE CENTER FOR INTELLIGENT DATA ANALYSIS), NETHERLANDS FORENSIC INSTITUTE

EXECUTIVE SUMMARY

Authorship verification is one subfield of authorship analysis. However, the majority of the research in the field of authorship analysis is on the authorship identification problem. The authorship verification problem has received less attention than the authorship identification problem. Thus, there is a demand for a study on the authorship verification problem.

The authorship verification problem of digital documents is becoming increasingly important as the criminals or terrorist organizations take advantage of the anonymity of the cyberspace to avoid being punished. Thus, it is critical for forensic linguistic experts to come up with effective methods to verify a short text written by a suspect. This master thesis project provides an exploratory study on the authorship verification models to solve the authorship verification problem.

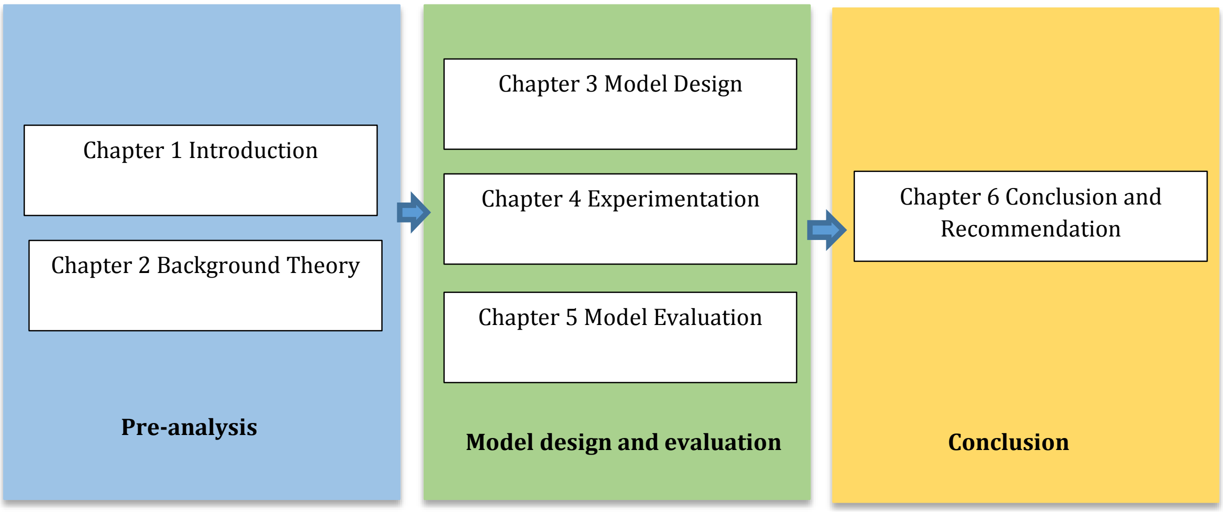
The research problem is as follows:

Given a few texts (around 1000 words each) of one author, determine whether the text in dispute is written by the same author.

The primary objective of this research is to design several innovative authorship verification models to solve the problem described above. A second goal of this research is to participate in the PAN Contest 2013 in the task of authorship verification.

This thesis project explores extensively the possibilities of using compression features to solve the authorship verification. Both one-class classification models and two-class classification models are designed in this project. In a one-class classification model, there is only target class, and the decision is based on a predefined rule. In a two-class classification model, there are both target class and outlier class, and the threshold is decided by learning the boundary between the two classes. In total five models have been designed and evaluated, four of which use compression features. Character N-Gram Model is designed in this research to make a comparison of character-grams and compression features.

The initial task of this project is the data collection. In order to participate in the PAN Contest, similar data (engineering textbooks from bookboon.com) were collected. In total 72 books written by 51 authors are in the collected corpus. The Book Collection Corpus was derived from the collected book and was used to develop the models. Additionally, an Enron Email Corpus was used to test the performance of one authorship verification model. As a result, the models designed received desirable performances and have shown potential to solve other similar problems. The design of the thesis report is as follows:


Chapter 1 Introduction
Chapter 2 Background Theory
Pre-analysis
Chapter 3 Model Design
Chapter 4 Experimentation
Chapter 5 Model Evaluation
Model design and evaluation
Chapter 6 Conclusion and Recommendation
Conclusion

Chapter 1 together with Chapter 2 is known as the pre-analysis stage. Chapter 1 describes the research problem, the research design as well as the research objectives; Chapter 2 explains the fundamental theories to design an authorship verification model. Chapter 3, Chapter 4 and Chapter 5 elaborates on the model design, model experiment results as well as model evaluation results. Finally, based on the study from Chapter 3, Chapter 4 and Chapter 5, conclusions have been reached and recommendations have been suggested in the Chapter 6.

ACKNOWLEDGEMENT

This thesis report concludes my final project of the master program *Management of Technology* and my internship at Knowledge and Expertise Center for Intelligent Data Analysis (KECIDA) at the Netherlands Forensic Institute. After one year and a half study, I have developed strong interests towards intelligent data analysis. Therefore, I wanted a thesis project where I could dive further in this field. First of all, I would like express my gratitude to my first supervisor Jan van den Berg, who has trusted my capabilities from the beginning and helped me to find the internship at KECIDA. As my external supervisor, Cor Veenman has given continuous support for my research. Without his support as well as criticism, this research would have not been finished in a high quality. Cor helped me go through the frustrating moments of my thesis project. Special thankfulness goes to Maarten Franssen, a philosopher who has acted as my second supervisor. Though Maarten has little domain knowledge, he spent two hours during our mid-term meeting to understand my work and give me valuable feedbacks. My experience of the thesis project supports a famous saying of Winston ChurChill '*Continuous effort –not strength or intelligence – is the key to unlocking our potential.*'

Additionally, I would like to thank the Faculty of Technology, Policy and Management, who awarded me the full scholarship to study this master program. Without the scholarship, my study in the Netherlands would have been almost impossible. Moreover, I would like to express my thankfulness to my family in China and South Korea. Whatever I decide, they fully support. It is their unconditional support and fully trust that makes it possible for me to strive for my own dream. Finally, my appreciation goes to my beloved friends in the Netherlands and in China. I couldn't imagine my life without their support and encouragement.

The Hague,

July, 2013

Zhenshi Li

Key Terms

Text: a digital file stored in the computer with the filename extension .txt.

Known texts: texts that are written by the same author. 'Known' means that the authorship is known.

Unknown text: one text that needs to be verified if it is written by the same author who wrote the known texts. Unknown indicates that the authorship is unknown.

Problem: a simple case that is to be solved, which consists of a few known texts from one author and one single unknown text that needs to be labeled/classified.

Profile: author profiling is one sub-field of authorship analysis. Nonetheless, profile-based approach is a type of data sampling approach.

UTF8: Universal Character Set Transformation Format-8 bit is designed to encode the characters with the Unicode 8 bit. This is selected as the encoding standard of all the texts in this research.

Book Collection Corpus: the well-prepared book corpus derived from the collected books. The books are from the bookboon.com, which is a website providing free textbooks for students. In total 51 authors with 72 books are in the collection.

Dataset V: a dataset derived from the Book Collection Corpus to test the performance of the designed models that use character *n*-gram features.

Dataset R: a dataset derived from the Book Collection Corpus to test the performance of the designed models that use compression features.

Precision: a performance measure derived from Information Retrieval, which evaluates the correctly predicted positive labels to all the predicted positive labels.

Recall: a performance measure derived from Information Retrieval, which evaluates the ratio of the correctly predicted positive labels to all the positive labels.

F-measure: a performance measure which aims to balance the precision and recall since the improvement of precision can be traded off by the decrease of recall.

AUC: a performance measure called Area under the Curve, which accompanies the Receiver Operating Characteristic in the Signal Detection Theory. The ratio can be interpreted as the accuracy of a predictive model.

Compression distance: distance of two texts measuring the length difference of the compressed texts is called compression distance. Several measures are proposed to indicate the compression distance between two texts.

Compression Algorithm: algorithms that compress files are called compression algorithm, such as 7zip, RAR etc.

COMPRESSION DISTANCE MEASURES: Formulas to measure the compression distance between two texts are called compression distance measures.

***NCD*:** Normalized Compression Distance, one type of compression distance measure.

***CLM*:** Chen-Li Metric, one type of compression measures.

***CosS*:** Compression-based Cosine metric.

***CDM*:** Compression-based Dissimilarity Method.

PPMD: PPM is short for Prediction by Partial Matching. PPMd is a variant of PPM.

PROTOTYPES: a subset of a dataset or a subset of a class, which are believed to be sufficient to represent the entire dataset or class.

TARGET CLASS: a class that consists of all the known documents from one author.

OUTLIER CLASS: though outlier more often refers to a very small number of objects, in this research it has the similar meaning as the others or the rest of the world.

INTER-COMPRESSION DISTANCES: compression distances between texts written by the same author.

INTRA-COMPRESSION DISTANCES: compression distances between texts written by different authors.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

One time I received an email from one of my friends, claiming that she was robbed in Paris during her short trip and was in desperate need of money. When I was reading the email, I had the feeling that it was not her who wrote it based on my knowledge of her. From the disputed email, I felt that it was a single girl who went to Paris alone secretly and had a terrible experience, while my friend is actually a married lady with two kids, which makes it unlikely that she would go on a trip by herself without telling her family. This is a simple real-life example of authorship analysis in the context of cyberspace.

1.1 CYBERSPACE AND CYBERCRIME

Cyberspace is the context of the authorship problems over the digital documents. According to the definition of the Department of the Defense (DoD) of the United States, cyberspace is the "the notional environment in which digitalized information is communicated over computer networks" (Andress & Winterfeld, 2011). The definition of cyberspace from United Nations (UN) is "the global system of inter-connected computer, communication infrastructures, online conferencing entities, databases and information utilities generally known as the Net." (Andress & Winterfeld, 2011). Thus, cybercrime can be literally interpreted as the crimes that are commited in the cyberspace.

1.2 AUTHORSHIP ANALYSIS

The problem of document authorship can be dated back to the medieval epoch (Koppel, Schler, & Argamon, 2009). As long as there is a written document, there might be a dispute over the authorship, for instance multiple people would claim the authorship of one piece of famous writing. The authorship study on the disputed Federalist Papers by Mosteller and Wallace (Mosteller & Wallace, 1963) is regarded as a contemporary hallmark. The rapid development of Information Technology, especially the Web Technology, has created an anonymous environment for the criminals or terrorist organizations to communicate or distribute illegal products (e.g. pirated software and stolen materials) via the internet (Zheng et al., 2006). It is said that criminals attempt to hide their personal information when sending online messages. Therefore the anonymity characteristic of online messages is a big challenge for the forensic experts to deal with the cybercrimes (Zheng et al., 2006). Consequently, one of the tasks of the forensic lingustic experts is to track the texts written by the same author but under different names (Kourtis & Stamatatos, 2011). Document authorship analysis has increasingly attracted the focus recently owing to the applications of the forensic analysis, the booming of the electronic transactions, development of humanities as well as the improvement of the computation methods (Koppel, Schler, & Argamon, 2009).

Analyzing the characteristics of a piece of written document to reach a conclusion on its authorship is called authorship analysis, the root of which is stylometry, a linguistic research field that utilizes the knowledge of statistics as well as machine learning techniques (Zheng et al., 2006). According to Zheng et al. (2006) there are three subfields that researchers focus on in the field of authorship analysis, i.e. authorship identification (authorship attribution), authorship profiling and similarity detection. Similarly, Koppel et al.(2009) summarized the problems that authorship analysis aims to solve into three scenarios, authorship identification (also called needle-in-a-haystack problem due to large size of candidate authors), authorship verification problem, and author profiling problem.

Authorship Identification: who wrote the document in dispute?

As is shown in Figure 1, the objective of authorship identification/authorship attribution is to determine the likelihood of a candidate author that wrote a text, i.e. which candidate is the most likely author of the disputed text. Given a text *d* and a set of candidate authors $C=\{a_1, a_2 ... a_n\}$, matching the author from set *C* with the text in dispute *d* is the problem to solve in this scenario. If the author of text *d* is definitely in the set *C*, this is called a closed set problem; if the author of text *d* can also be someone who is not included in the set *C*, this is well-known as an open set problem.

The rationale of authorship identification is:

Different authors have different writing styles, based on which one author can be distinguished from the others.

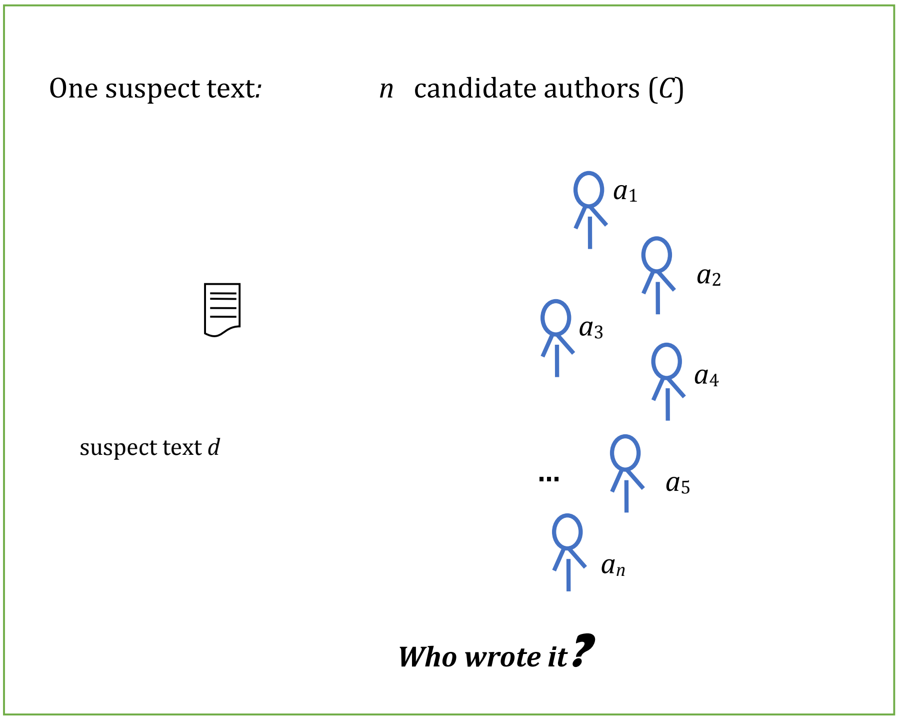


FIGURE 1: AUTHORSHIP IDENTIFICATION

Authorship Verification: Did the suspect author write the text in dispute?

The task of authorship verification is to assess whether the text in dispute was written by the same author of the known texts. Given a suspicious text d and a set of known texts from one author A= $\{d_1, d_2 ...d_n\}$. The problem is to verify whether the text d in dispute was written by the same author who wrote the known texts from the set A.

The underlying rationale of authorship verification is:

The same author has some writing styles that are believed to be difficult to camouflage in a short period of time, and therefore based on these writing styles a document claimed from the same author can be verified.

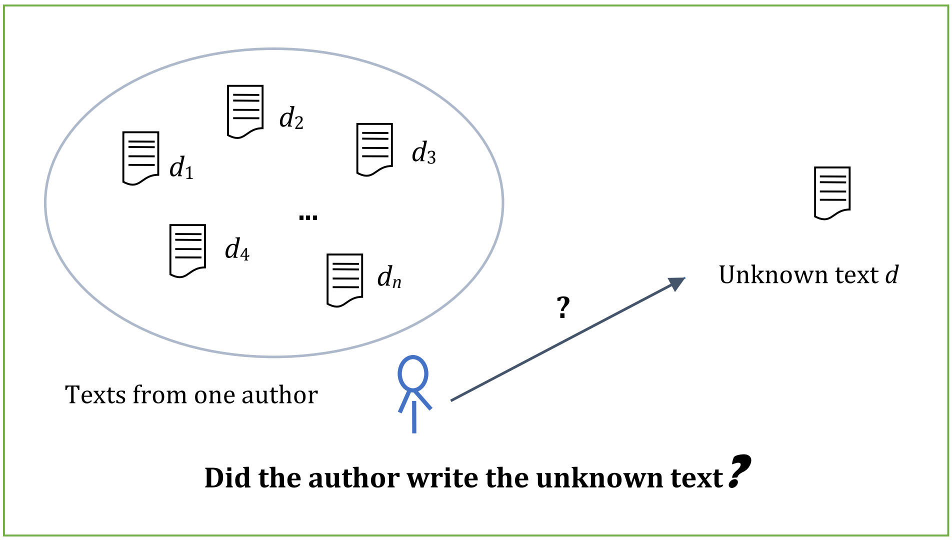


FIGURE 2: THE AUTHORSHIP VERIFICATION

Different from the authorship identification problem, there are no candidate authors in this scenario but there is one suspect author who is known to have written all the known texts. The only study objects in this scenario are the known texts from the set A, and the unknown text d which will be labeled with *YES* or *NO*.

Author Profiling: What are the characteristics of the author of the document in dispute?

Regarding to the number of authors, there are many candidate authors in the authorship identification problem, whereas in the author profiling problem there is one author and the study is to generate a demographic profile (for instance gender, age, native language) of the author based on the given documents (Koppel et al., 2009).

The underlying rationale of author profiling is:

Authors' written documents can reveal some of their personal characteristics without their consciousness, based on which a demographic author profile can be created.

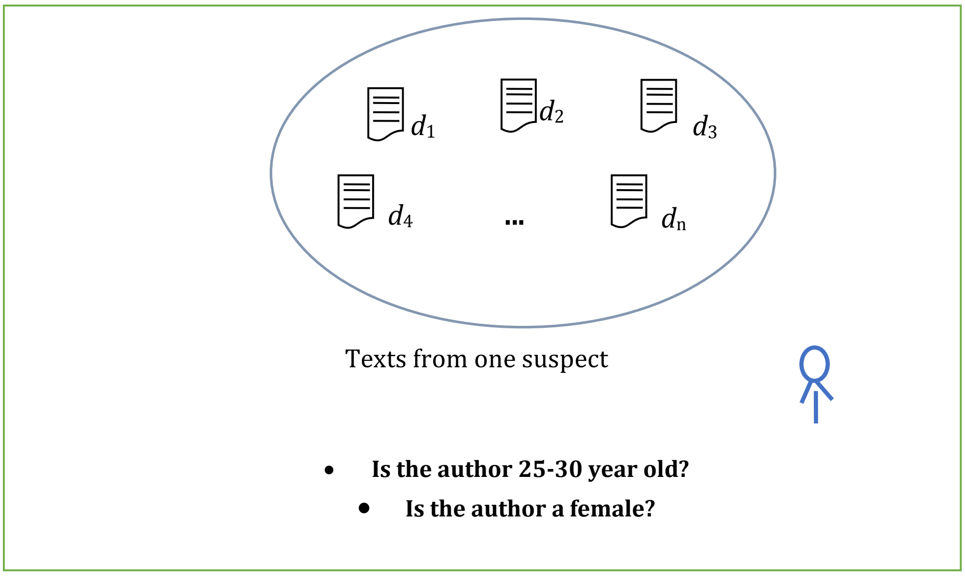


FIGURE 3: AUTHOR PROFILING

Some researchers (Koppel et al., 2009; Juola, 2006) regard authorship attribution equal to authorship identification and treat authorship verification and author profiling as variations of the authorship identification problem. The summary of the definition of the authorship analysis categories is described in Table 1. In fact, authorship identification is also regarded as a multiclass classification, and authorship verification is well-known as a binary classification problem (Mikros & Perifanos, 2011). Nevertheless, text classification does not apply to author profiling. In this sense, author profiling is rather different compared with authorship identification and authorship verification.

TABLE 1: SUBFIELDS OF AUTHORSHIP ANALYSIS (ZHENG, LI, CHEN, & HUANG, 2006)

Field category	Description
Authorship Identification/ Authorship Attribution	Study a text in dispute and find the corresponding author in a set of candidate authors.
Authorship Verification/ Similarity Detection	Compare multiple pieces of writing and determine whether they are written by the same author without identifying the author.
Authorship Characterization/ Author Profiling	Detect unique characteristics of an author's written texts and create an author profile.

Many researchers who study authorship attribution focus on a small size of authors and tend to study large size documents for instance using documents over 10,000 words, which is regarded as an artificial situation (Luyckx & Daelemans, 2008). It is argued that they prefer to conduct the research in this way to get a better performance of their study, resulting in an overestimated outcome of their designed models (Luyckx & Daelemans, 2008). However, the most urgent problems the forensic experts face are related to the computer-mediated communication documents (e.g. emails, blogs), which are usually short in terms of the document length and have a large size of potential candidates, the amount of which are usually unknown. In many cases, they need to determine that whether several documents, such as emails, are written by the same author without identifying the real author.

1.2 PROBLEM DESCRIPTION OF THE RESEARCH

In the context of cyberspace, a digital document found can be used as an evidence to prove that a suspect is a criminal if he/she is the author of the document. If the suspect authors are unknown i.e. there is no suspect, thus this is commonly known as an authorship identification problem. However, there are also some cases when the identification of the author is not necessary, i.e. it is enough just to know if the document in dispute was written by the author of the documents that are given. This is a problem faced by many forensic linguistic experts which is called as authorship verification problem. Derived from the problem description, the main research question is described as follows.

Research problem statement

A few texts each around 1000 words from one identified author are given, and there is one single text whose authorship is unknown. The problem is if the author of the given known texts is also the author of the unknown text.

This thesis project will design a classification model to determine whether a suspect has written a document or not, i.e. the result is in the set of L= {YES, NO}.

1.3 Overview of Data mining Process

Text mining is one variant of data mining, and thus the basic data mining process also applies to text mining as well as this research. A framework of the data mining process called Cross-Industry Standard Process for Data Mining is widely adopted by various industries (Olson & Delen, 2008).

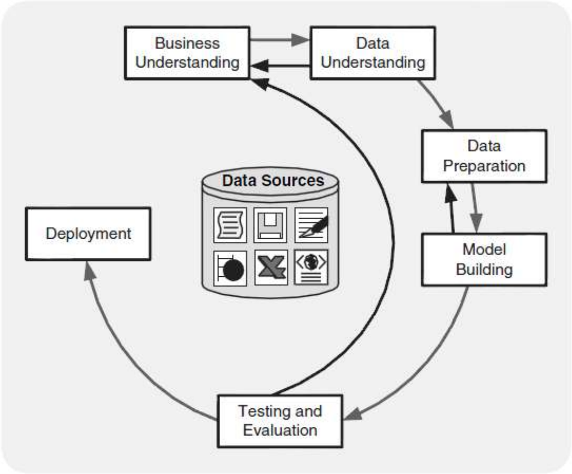


FIGURE 4: CRISP-DM PROCESS (OLSON & DELEN, 2008)

The structure of this research is derived from the CRISP-DM process, which is shown in Figure 4. The *Business Understanding* step is on essence understanding the underlying problem and the objective. *Data understanding* step solves the problem of targeting the right data sources and sampling the appropriate data, and selecting the proper features. Further, *Data Preparation* step, which is also known as data pre-processing step, is designed to further cleanse the data. In the step of *Model Building*, two datasets are used: *training dataset* and *validation dataset*. The *training dataset* aims to train the model, and the *validation dataset* is used to tune the corresponding parameters. In the last step *Testing and Evaluation*, testing task of this step involves the *test dataset*, and evaluation task is designed to check the alignment of the designed model with the original defined problem.

1.4 Research Objective

The objective of this research is to creatively design several authorship verification models and reach a relatively high performance. A second goal of this research is to participate in the PAN Contest 2013 in the task of authorship verification. The deliverables of the research are the designed authorship verification models.

1.5 Research Scope

This project is part of the research at KECIDA (Knowledge and Expertise Centre for Intelligent Data Analysis) from the Netherlands Forensic Institute and is applicable to participate in the PAN 2013 Contest in the track of authorship verification. This study focuses on the English language, i.e. a study on the other languages is out of the scope.

Moreover, Matlab and its Pattern Recognition toolbox (Duin et al., 2007) developed by TU Delft are the main tools in this study.

1.6 RESEARCH QUESTIONS

Three research questions are generated in alignment with the research objective and the research framework. The research can be divided into three phases, *Pre-analysis phase*, *Model Design phase* and *Model Evaluation phase*. Derived from the data mining process in the Figure 4, the tasks in different phases are planned as the Figure 5.

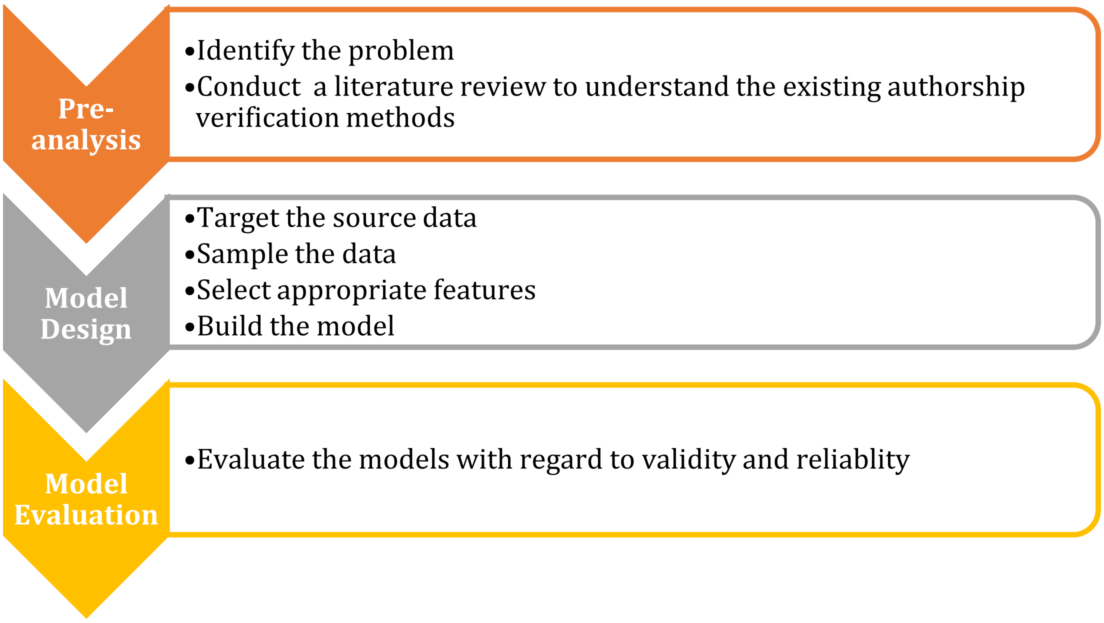

Pre-analysis
•Identify the problem
•Conduct a literature review to understand the existing authorship verification methods
Model Design
•Target the source data
•Sample the data
•Select appropriate features
•Build the model
Model Evaluation
•Evaluate the models with regard to validity and reliablity

FIGURE 5: THREE-PHASE RESEARCH

According to the tasks in the Figure 5, research questions are formulated as follows:

Q1: What are the existing methods that have been used to solve the similar problem?

A thorough understanding of the literature in this field is critical to have a good outcome. This question is designed to give an overview of the existing research. The literature study will focus on the features and computation techniques.

Q2: How the model should be designed?

This is the crucial part of the research. The design of the model can be operationalized into the following steps.

- Targeting the source data
- Data sampling
- Feature extraction and selection
- Computation technique selection

Q3: How is the validity of the designed model?

This question deals with the Model Evaluation phase. It is essential to see how valid and trustworthy a design research is. The designed models will be assessed with the test dataset to gain insights into the validity and reliability of the models.

CHAPTER 2 BACKGROUND THEORY

This chapter elaborates on the key background theories regarding to data sampling, data features, computation techniques, and the assessment measures.

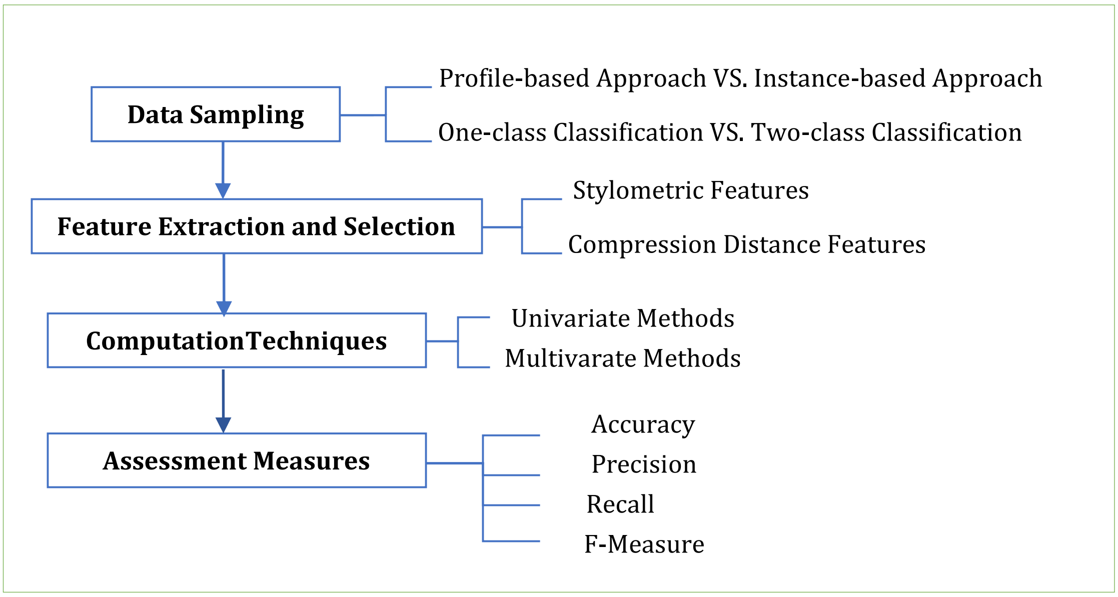

Data Sampling
Profile-based Approach VS. Instance-based Approach
One-class Classification VS. Two-class Classification
Feature Extraction and Selection
Stylometric Features
Compression Distance Features
ComputationTechniques
Univariate Methods
Multivarate Methods
Assessment Measures
Accuracy
Precision
Recall
F-Measure

FIGURE 6: OVERVIEW OF THE KEY CONCEPTS OF CHAPTER 2

2.1 PROFILE-BASED APPROACH VS INSTANCE-BASED APPROACH

There are various authorship attribution methods, and according to Stamatatos (2009), all the methods can be classified into two groups: profile-based approach and instance-based approach. Figure 7 describes the profile-based approach, which is a process of concatenating all the training texts of one author and generating an author profile. The features of each author are extracted from the concatenated text. Extracted features are used in the attribution model to determine the most likely author of the dispute text. However, a profile-based approach is criticized for losing much information because of the generating profile-based feature process which is required to remove all the dissimilar contents from the same author.

On the contrary, instance-based approach, which is used in most of the contemporary authorship attribution research, can keep most of the information from the given texts, and extracted features are applied to a machine learning classifier. The procedures of the instance-based approach can be seen in the Figure 8.

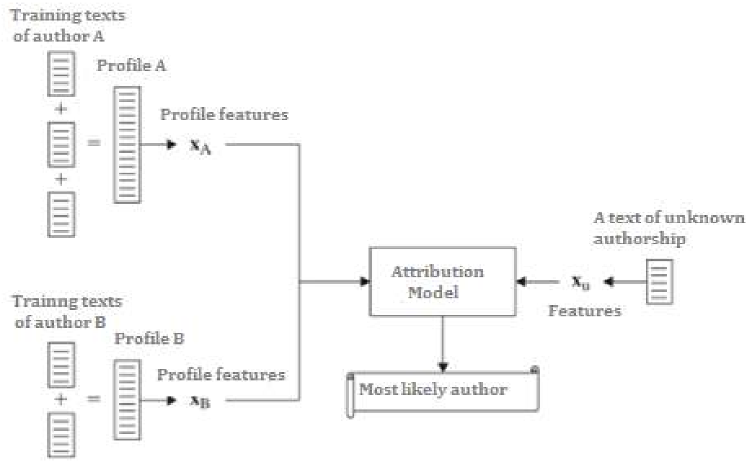


Figure 7: Profile-based approaches (Stamatatos, 2009)

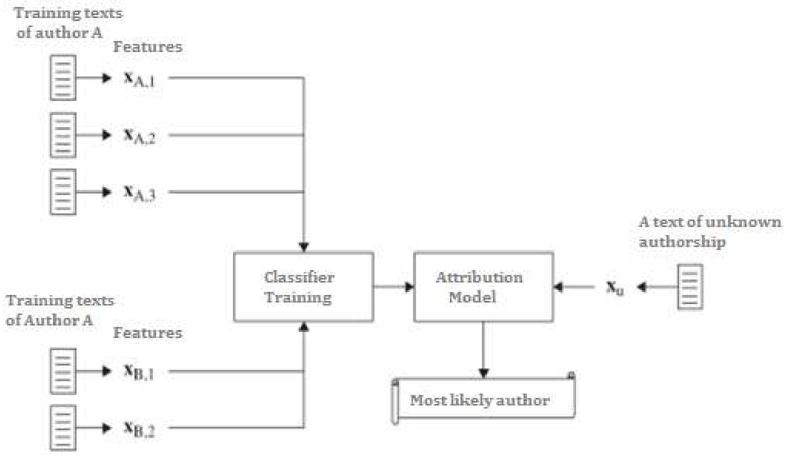


Figure 8: Instance-based approaches (Stamatatos, 2009)

2.2 Two-class Classification VS One-class Classification

Researchers have used both two-class classification methods, and one-class classification methods (Koppel & Schler, 2004). It depends on the authorship analysis task to decide on the appropriate data sampling approach.

2.2.1 One-class Classification

One-class classification can be literally interpreted as there is one class only. Thus the result is simply that the studied object is in the class or not. The one-class classification description is shown in Figure 9. The existing class is the texts at hand in this research, which is known as the target class. What is the problem to be solved is the likelihood of the unknown text is drafted by the same author. InFigure 9, it seems that there are two classes, the target class and the outlier class. However the diversity of the outlier class is huge. This can be easily explained with an example from real life.

One-class classification example:

Given an unknown person, we would like to know that the probability that the person is Chinese. In this example, people will determine based on their knowledge of Chinese people's features (their common appearances, the way they behave etc.) and theoretically no one will try to generalize the features that non-Chinese have, since the study to classify non-Chinese is not realistic and feasible.

This is also the case in this research, i.e. the complete study of the outlier class is almost a mission impossible and ineffective to solve the problem. Therefore, authorship verification research features strongly with the one-class classification's characteristics.

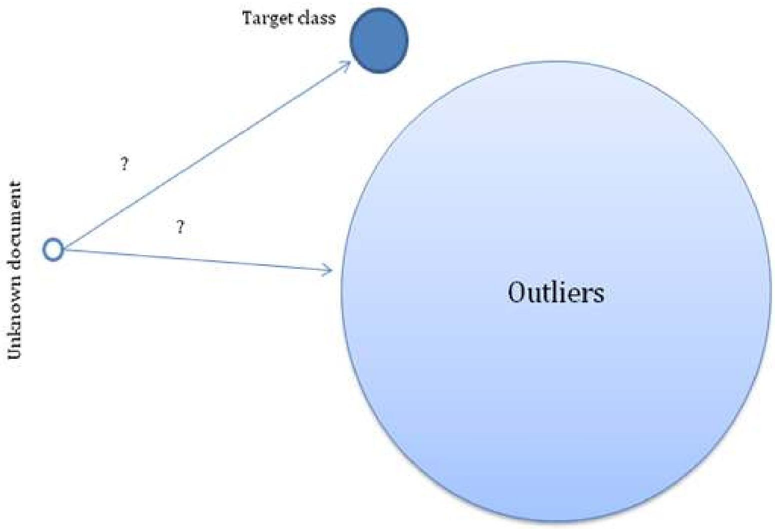


FIGURE 9: ONE-CLASS CLASSIFICATION DESCRIPTION

2.2.2 TWO-CLASS CLASSIFICATION

It seems that the authorship verification problem features strongly with one-class classification characteristics. Nevertheless, when the texts from the author are rather limited, the one-class approach can be highly biased and consequently the result is not reliable. Under this circumstance, the outlier class can be created artificially for machine learning classifiers to learn to discriminate the two classes. Figure 10 describes the two-classification approach with appropriate outlier selection. The selection of the proper outlier representation is of great importance, which is supposed to be as close to the target as possible.

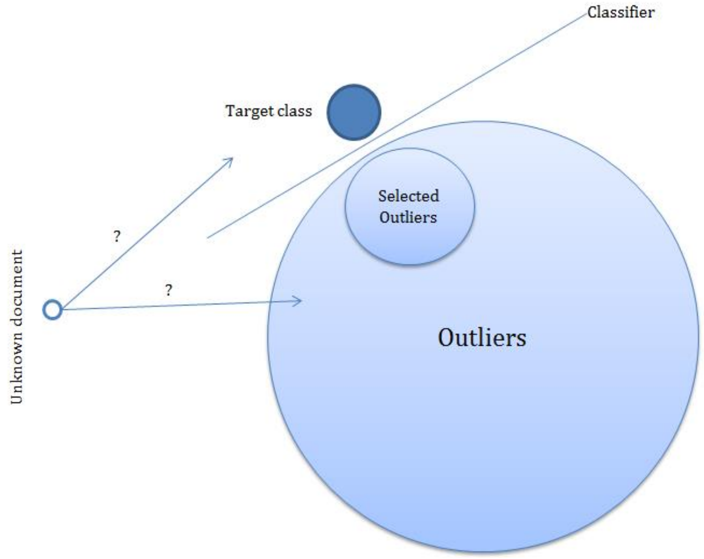


FIGURE 10: TWO-CLASS CLASSIFICATION WITH THE APPROPRIATE SELECTED OUTLIERS

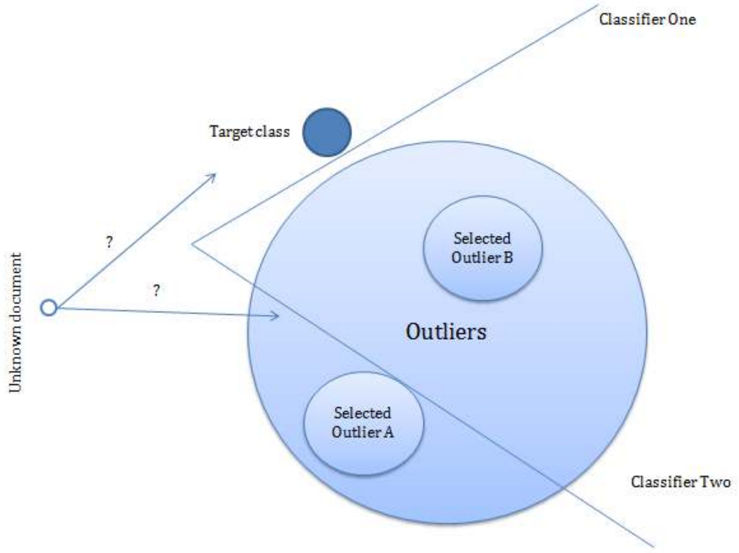


FIGURE 11: TWO-CLASS CLASSIFICATION WITH INAPPROPRIATE SELECTED OUTLIER A

Figure 11 describes the two-class classification with inappropriate selected outliers. It shows that if the selected outliers are far away from the target class such as *Selected*

Outlier A, then any classifier between *Classifier One* and *Classifier Two* in Figure 11 are regarded as effective to classify between the target class and the Selected Outlier A. Consequently, the model designed based on *Selected Outlier A* is very likely to label other outliers that are closer to the target class, for instance all the cases from Selected Outlier B, as target class. Therefore, misclassification between target class and outliers is more likely.

Two-class classification example:

Given an unknown person from East Asia, we would like to know the probability that the person is Chinese. However, what we have already known are only two to five Chinese people, which makes it impossible for us to generate a feature-based profile to determine the features of Chinese. We need a comparison class or outlier class.

Thus, we know that Japanese and Koreans look similar to Chinese and luckily we have a lot of information about them, so we decide to select the features of Japanese and Korean people as the comparison class and determine whether the unknown person is Chinese or not based on the similarity between the person to the small Chinese class or to the selected outlier class.

2.3 FEATURES

The most traditional features in the study of authorship analysis are stylometric features, while some researchers such as Lambers and Veenman (2009) have attempted to use compression distances between texts as innovative features to approach the authorship verification problem. Therefore, in this section, both stylometric features and compression features will be explained.

2.3.1 STYLOMETRIC FEATURES

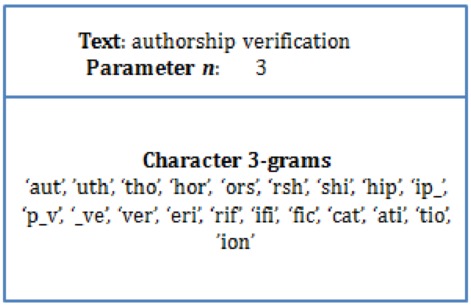
The stylometric features of documents are of great importance to determine the writing styles. Table 2 summarizes the seven different types of stylometric features, i.e. lexical features, character features, syntactic features, structural features, content-specific features, semantic features, and idiosyncratic features.

LEXICAL FEATURES

Lexical features, also known as word-based features/token-based features are language-independent, which means that they can be applied to almost all the languages with the assistance of a tokenizer, though the tokenizing workload of some languages (e.g. Chinese) is rather heavy (Stamatatos, 2009). Some effective lexical features are word length distribution, average number of word percentage as well as vocabulary richness (Iqbal, Fung, Khan, & Debbabi, 2010). Moreover, some researchers (Escalante, 2011; Mikros & Perifanos, 2011; Tanguy et al., 2011) have used word *n*-grams to solve the authorship attribution problems. However, richness of vocabulary is claimed that it might be ineffective because a great many word types from the texts are *hapax legomena,* which means they only appear once in the entire text. Therefore the difference between texts written by the same author can be as different as the texts written by different authors with regard to the vocabulary richness (HOOVER, 2003).

CHARACTER FEATURES

Features such as letter frequency, capital letter frequency, total number of characters per token and character count per sentence are regarded as the most powerful character features (Iqbal, Fung, Khan, & Debbabi, 2010). It is commonly believed that these features can imply the author's preference of using some special characters (Iqbal, Fung, Khan, & Debbabi, 2010). Moreover, character *n*-grams, which are consecutive sequences of *n* characters, have been proved to be effective to solve the topical similarity problems (Damashek, 1995). An example of character *n*-grams is shown in Figure 12. Additionally, the selection of parameter *n* of character *n*-gram featureshas significant impact on the result (Stamatatos, 2009). According to Stamatatos (2009), if the parameter *n* is small (e.g. 2, 3), then the character *n*-grams would be able to represent sub-word information such as syllable information, but it fails to capture the contextual information. If the *n* is large, it would be able to represent contextual information such as thematic information. The effect of the parameter *n* of the character *n*-grams is described in the Figure 13.



Text: authorship verification
Parameter *n*: 3

Character 3-grams
'aut', 'uth', 'tho', 'hor', 'ors', 'rsh', 'shi', 'hip', 'ip_', 'p_v', '_ve', 'ver', 'eri', 'rif', 'ifi', 'fic', 'cat', 'ati', 'tio', 'ion'

FIGURE 12: EXAMPLE OF CHARACTER N-GRAMS

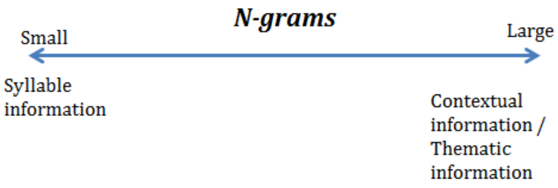


FIGURE 13: THE EFFECT OF THE PARAMETER N

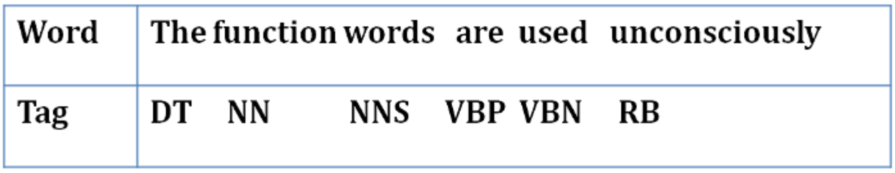
Additionally, Grieve (2007) conducted a research to evaluate 39 textual measurement techniques including word-length features, sentence-length features, vocabulary richness features, grapheme frequency features, word frequency features, punctuation mark frequency features, collocation frequency features and character-level *n*-gram frequency features. Grieve (2007) prepared the Telegraph Columnist Corpus with 1600 texts (with an average text length equals to 937 words in the range from 500 words to 2000 words) from 40 authors with similar backgrounds, and the results were compared based on the following CHI-squared statistic equation:

$$\chi^2 = \sum\left(\frac{(Oi-Ei)^2}{Ei}\right) \qquad i=1, 2, 3\ldots n \quad (2.1)$$

Where *Oi* represents the observed frequencies, and *Ei* represents the expected frequencies. Based on chi-squared statistic evaluation measure, Grieve (2007) found out that the most desirable results on the Telegraph Columnist Corpus were from word and punctuation mark combination, character 2-grams/bigrams and 3-grams/trigrams. The test accuracy of word and punctuation mark combination among 40 possible authors was 63%, while test accuracy of character bigrams and trigrams among 40 possible authors were 65% and 61% respectively (Grieve, 2007).

SYNTACTIC FEATURES

Baayen, van Halteren and Tweedie first discovered the effectiveness of syntactic elements (e.g. punctuation marks and function words) in identifying an author (Baayen, van Halteren, & Tweedie, 1996). The selection of function words is arbitrary and usually highly dependent on the language expertise (Stamatatos, 2009). Moreover, it is believed that the function words are used unconsciously and it is consistent by the same author regardless of the topics and has a low possibility of being deceived (Koppel, Schler, & Argamon, 2009). Additionally, many researchers have also adopted the frequencies of part-of-speech tagging as deterministic stylometric features (Koppel, Schler, & Argamon, 2009). One of the popular English part-of-speech tags are Upenn Treebank II Tags, the full list of which can be found in the Appendix A. Figure 14 illustrates an example of part-of-speech tagging.



Word	The function words are used unconsciously
Tag	DT NN NNS VBP VBN RB

FIGURE 14: PART-OF-SPEECH TAGGING EXAMPLE

STRUCTURAL FEATURES

The unit of analysis of structural features is the entire text document, and the structural features evaluate the overall appearance of the document's writing style (Iqbal, Fung, Khan, & Debbabi, 2010). The structural features commonly used are average paragraph length, number of paragraphs per document, presence of some structured layouts such as the place of the greetings and recipient's address in an email (Iqbal, Fung, Khan, & Debbabi, 2010). In the authorship verification problem of computer–mediated online messages such as blogs and emails the structural features may be very promising (Koppel, Schler, & Argamon, 2009).

CONTENT-SPECIFIC FEATURES

Content-specific features are dependent on the topics of the documents, which are a collection of the keywords in the specific topic domain (Iqbal, Fung, Khan, & Debbabi, 2010). In addition, the biggest disadvantage of the content-specific features is that they may vary substantially in different topics with the same author. Consequently, the high performance of one model using content-specific features may perform badly if the topic has changed (Koppel, Schler, & Argamon, 2009). For instance, the keywords of an article on financial crisis would be much different from the keywords of an article on cyber security. Therefore, the selection of the content-specific features is tailor-made to a specific context and should be dealt carefully.

SEMANTIC FEATURES

Semantic features are called as rich stylometric features compared with the poor features such as character *n*-grams (Tanguy et al., 2011). WordNet, which is a project from Princeton University, is a high quality source of word synonyms and hypernyms (Fellbaum, 1998). According to the result of Tanguy et al. (2011), simply depending on the rich stylometric features did not reach desirable results, nevertheless the combination of the rich features with poor features has improved the results obtained using them separately.

IDIOSYNCRATIC FEATURES

Idiosyncratic features refer to the presence of mistakes, e.g. spelling mistakes and syntactic mistakes in the document (Iqbal, Fung, Khan, & Debbabi, 2010). There is the correct spelling of a word. In terms of English, there are correct British spelling and correct American spelling. Hence, it is not difficult to assess whether a word is written correctly or not, whereas the number of incorrect forms of a word can be infinite. Thus, it is difficult to make a collection of all the idiosyncratic features, but it is possible to make a list for each person based on analysis on the spelling errors and syntactic errors from the existing written documents of the author.

TABLE 2: DIFFERENT TYPES OF STYLOMETRIC FEATURES

NO.	Stylometric features	Example features
1	**Lexical features**	Word *n*-grams, word length distribution, average number of words per sentence, vocabulary richness,
2	**Character features**	Frequency of letters, frequency of capital letters, total number of characters per token, character count per sentence, and character n-grams.
3	**Syntactic features**	Punctuation, function words (e.g. "upon", "who" and "above") and part-of-speech tagging.
4	**Structural features**	Average paragraph length, number of paragraph per document, presence of greetings and their position in the documents etc.
5	**Semantic features**	Synonyms, hypernyms etc.
6	**Content-specific features**	They are collections of keywords in a domain, and may be different in different contexts.
7	**Idiosyncratic features**	They are collections of common spelling mistakes and grammatical mistakes.

STYLOMETRIC FEATURE SELECTION

Automatic feature selection is a process of removing non-informative features (Yang & Perdersen, 1997). Some traditional features selection methods in text categorization are Document Frequency (DF), Information Gain (IG), Mutual Information (MI), a χ^2-test (CHI) etc. Yang & Perdersen (1997) found that IG and CHI were the most effective methods in their experiments of text categorization. Additionally, strong correlations were found between DF, IG and CHI when valuing the same term, indicating that the low-cost method DF can be reliable as well (Yang & Perdersen, 1997).

- Document Frequency (DF)

Document Frequency records the number of times a term occurs in the documents from one category. A predetermined threshold was set to exclude the low frequent terms, and the rationale of the threshold setting is that rare terms are not informative (Yang & Perdersen, 1997).

- Information Gain (IG)

According to Kanaris et al.(2007), information gain of a term can be formulated as follows:

$$IG(C,t) = H(C) - H(C|t) \qquad (2.2)$$

Where C is the class/category, $H(C)$ is the entropy of the C, t is one term that is present in the class C, and $H(C|t)$ is the entropy of C when t is present. If $IG(C,t)$ is zero, it indicates that the presence of t has no impact on differentiating the class C from other classes. If $IG(C,t)$ is approaching one, it implies that t is one distinctive feature of the class C.

- Mutual Information (MI)

TABLE 3: CONTINGENCY TABLE OF TERM T AND CLASS C

	Term t occurs (number of documents)	Term t does not occur (number of documents)
In the class C	a	c
Not in the class C	b	d

In Table 3, a denotes the number of documents from the class C that have the term t, b denotes the number of documents that have the term t but are not from the class C, c denotes the number of documents in the class C that do not have the term t, and d denotes the number of documents that do not have the term t and are not in the class C. The mutual information is formulated as follows (Yang & Perdersen, 1997):

$$I(t,C) = log\frac{P_r(t \wedge C)}{P_r(t) \times P_r(C)} \qquad (2.3)$$

and $I(t,c)$ can be estimated with the formula in equation 2.4:

$$I(t,C) \approx log\frac{a \times n}{(a+c) \times (a+b)} \qquad (2.4)$$

Where n denotes the number of all the documents, i.e. $n=a+b+c+d$.

- χ^2 statistic (CHI)

The χ^2 statistic of the term t and the class C which has one degree of freedom measures the independency of the term t and the class C. It can be formulated as equation 2.5based on the equation 2.1 and the Table 3. If $\chi^2(t,C)$ is zero, it implies that the term t and the class C are independent.

$$\chi^2(t,C) = \frac{n \times (ad-cb)^2}{(a+c) \times (a+b) \times (b+d) \times (c+d)} \qquad (2.5)$$

In the equation 2.5, n denotes the number of all the documents.

It is said that authorship attribution and text categorization are quite related. For instance, a closed set authorship attribution problem can be regarded as a multi-class text categorization problem (Stamatatos, 2009). Nonetheless, authorship attribution values the style of writing, while text categorization focuses only on the text content (Bozkurt et al., 2007). As a consequence, not all the methods mentioned above can be effectively applied to select relevant stylometric features for authorship attribution.

In fact, stylometric experts usually examine a text and select some stylometric features manually. A popular criterion for the feature selection is based on the frequencies of the stylistic features (Stamatatos, 2009). Orsyth and Holmes (1996) compared a feature set selected by frequencies with a feature set selected by the distinctiveness and they found that the features selected by distincitiveness were more accurate in their experiment. Another comparison of features selected by frequencies with the features selected by the infomation gain showed that features selected based on frequencies were more accurate (Stamatatos, 2009).

2.3.2 COMPRESSION FEATURES

Compression features refer to the compression distances scaled by a specific compression distance measure. Compression distance features can be applied to both profile-based approach and instance-based approach, while it is claimed by Stamatatos (2009) that based on a review of the relevant literature, using compression features is more successful in combination with the profile–based approach. With regard to the compression models, the selection of the most appropriate compression algorithm as well as the compression distance measure is of great importance.

KOLMOGOROV COMPLEXITY

Kolmogorov complexity measures the minimum computational resources of an object (e.g. file) that are sufficient to reproduce the object. With regard to strings, Kolmogorov complexity is the length of the shortest description of the strings in a universal description language. Kolmogorov complexity was introduced and motivated by Solomonoff (1964), Kolmogorov (1965) and G. Chaitin (1969) independently. A simple example is described in Table 4. String A which has 66 characters can be described with string A', which only has 11 characters. If string A' is the shortest description of all the possible descriptions of string A, then 11 is the Kolmogorov complexity of string A.

TABLE 4: A SIMPLE EXAMPLE OF KOLMOGOROV COMPLEXITY

String A	'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaa'	66 characters
Short description A'	'a 66 times'	11 characters

COMPRESSION MEASURES

A compression-based dissimilarity method which is based on Kolmogorov complexity is proposed by Keogh et al. (2004) and is employed by Lambers and Veenman (2009) to solve forensic authorship attribution problem. The *Compression-based Dissimilarity Method* (*CDM*) is defined by Keogh et al. (2004) as follows:

$$CDM(x,y) = \frac{C(xy)}{C(x) + C(y)} \qquad (2.6)$$

Where C is the compression algorithm, $C(x)$ is the compressed length of the compressed document) of object x, $C(y)$ is the compressed length of object y, and $C(xy)$ is the compressed size of the concatenated object xy. When x and y are the same, then CDM (x, y) is close to 0.5, and when x and y are completely different, and then CDM (x, y) is approaching 1. An example of this measure is shown in Table 5. $Text_1$ to $Text_{10}$ are 10 different texts with similar text lengths. Among them, $Text_1$ and $Text_9$ are from the same author, $Text_6$ and $Text_7$ are from the same author, and $Text_3$ and $Text_4$ are from the same author. Compression distances from the same author are colored red. From the Table 5, it can be seem that the lowest values are from the diagonal line, which are the compression distances from the texts to themselves. The values in red are smaller than the values in black (except the one on the diagonal line) from the same row or column. This indicates that the compression distances between texts written by the same author are smaller than the compression distances between texts written by different authors. This is a plausible indication. Roger McHaney is the author of $Text_1$ and $Text_9$, 31 texts with the similar length of the *Text*$_1$ and *Text*$_9$ are selected to calculate the intra-compression distances. Every value in Table 6 represents the maximum distance from one text to the others. Therefore, in total 31 values sorted in an ascending order are shown in Table 6. Some values (e.g. 0.9460) in Table 6 are higher than a few values (e.g. 0.9395) in Table 5, which implies that *CDM* intra-compression distances sometimes are larger than *CDM* inter-compression distances.

Table 5: CDM Matrix of Ten Texts (Compression Algorithm: PPMd; Blue represents Diagonal values; Red represents distances from the same author)

CDM	*Text*$_1$	*Text*$_2$	*Text*$_3$	*Text*$_4$	*Text*$_5$	*Text*$_6$	*Text*$_7$	*Text*$_8$	*Text*$_9$	*Text*$_{10}$
Text$_1$	0,6866	0,9342	0,9440	0,9475	0,9386	0,9367	0,9394	0,9360	0,9287	0,9418
Text$_2$	0,9357	0,6830	0,9525	0,9598	0,9411	0,9466	0,9522	0,9180	0,9455	0,9535
Text$_3$	0,9426	0,9498	0,6684	0,8971	0,9519	0,9365	0,9359	0,9422	0,9530	0,9325
Text$_4$	0,9480	0,9582	0,8978	0,6598	0,9513	0,9400	0,9328	0,9492	0,9603	0,9387
Text$_5$	0,9425	0,9416	0,9540	0,9540	0,6917	0,9567	0,9526	0,9448	0,9506	0,9561
Text$_6$	0,9367	0,9448	0,9341	0,9393	0,9533	0,6692	0,9274	0,9390	0,9502	0,9422
Text$_7$	0,9353	0,9485	0,9345	0,9304	0,9466	0,9250	0,6641	0,9461	0,9564	0,9329
Text$_8$	0,9378	0,9153	0,9446	0,9526	0,9443	0,9395	0,9486	0,6788	0,9544	0,9502
Text$_9$	0,9281	0,9417	0,9518	0,9568	0,9463	0,9497	0,9569	0,9503	0,6694	0,9498
Text$_{10}$	0,9414	0,9495	0,9315	0,9369	0,9479	0,9410	0,9338	0,9471	0,9496	0,6671

Table 6: Summary of the maximum distances of texts from one author (Roger McHaney)

CDM	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6
Row 1	0.9292	0.9310	0.9321	0.9324	0.9329	0.9334
Row 2	0.9334	0.9342	0.9343	0.9345	0.9348	0.9352
Row 3	0.9354	0.9355	0.9358	0.9360	0.9376	0.9376
Row 4	0.9376	0.9379	0.9385	0.9387	0.9391	0.9393
Row 5	0.9394	0.9395	0.9416	0.9419	0.9426	0.9460
Row 6	0.9460					

Compression-based dissimilarity method that is co-developed by Li et al. (2004) and Cilibrasi and Vitányi (2005) is called the *Normalized Compression Distance* (*NCD*). The definition of the Normalized Compression Distance is as follows:

$$NCD(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}} \qquad (2.7)$$

Where $C(xy)$ is the compressed size of the concatenated object xy, $C(x)$ is the compressed result of object x, and $C(y)$ is the compressed size of object$y$. According to Cilibrasi and Vitányi (2005), within a certain boudary which requires that $C(xx)$= $C(x)$, th result of the normalized compression distance falls in the boundary [0,1+ε], where ε is the error. Table 7 describes *NCD* measure with the same 10 texts that were used in the *CDM*.Among them, *Text*$_1$ and *Text*$_9$ are from the same author, *Text*$_6$ and *Text*$_7$ are from the same author, and *Text*$_3$ and *Text*$_4$ are from the same author. Compression distances from the same author are colored red. From the Table 7, it can be seem that the lowest values are from the diagonal line, which are the compression distances from the texts to themselves. The values in red are smaller than the values in black (except the one on the diagonal line) from the same row or column. Table 8 illustrates the maximum intra-compression distances of the 31 texts from the same author (Roger McHaney) who wrote the *Text*$_1$ and *Text*$_9$. Therefore, in total 31 values sorted in an ascending order are shown in Table 8. Some values (e.g. 0.9035) in Table 8 are higher than a few values (e.g. 0.8740) in the Table 7, which indicates that *NCD* intra-compression distances sometimes are larger than *NCD* inter-compression distances.

TABLE 7: NCD MATRIX OF TEN TEXTS (COMPRESSION ALGORITHM: PPMD; BLUE REPRESENTS DIAGONAL VALUES; RED REPRESENTS DISTANCES FROM THE SAME AUTHOR)

NCD	*Text* $_1$	*Text* $_2$	*Text* $_3$	*Text* $_4$	*Text* $_5$	*Text* $_6$	*Text* $_7$	*Text* $_8$	*Text* $_9$	*Text* $_{10}$
***Text* $_1$**	0,3732	0,8697	0,8970	0,9036	0,8892	0,8740	0,8792	0,8758	0,8588	0,8892
***Text* $_2$**	0,8727	0,3660	0,9118	0,9256	0,8928	0,8937	0,9050	0,8392	0,8911	0,9105
***Text* $_3$**	0,8944	0,9070	0,3368	0,7943	0,9057	0,8827	0,8818	0,8908	0,9128	0,8699
***Text* $_4$**	0,9044	0,9225	0,7959	0,3197	0,9045	0,8893	0,8761	0,9041	0,9265	0,8819
***Text* $_5$**	0,8962	0,8937	0,9098	0,9097	0,3835	0,9214	0,9142	0,8977	0,9101	0,9169
***Text* $_6$**	0,8740	0,8902	0,8783	0,8880	0,9152	0,3383	0,8551	0,8809	0,9007	0,8893
***Text* $_7$**	0,8710	0,8976	0,8792	0,8718	0,9033	0,8503	0,3282	0,8950	0,9133	0,8718
***Text* $_8$**	0,8792	0,8339	0,8954	0,9106	0,8968	0,8818	0,8998	0,3576	0,9105	0,9023
***Text* $_9$**	0,8575	0,8835	0,9105	0,9198	0,9021	0,8998	0,9142	0,9026	0,3388	0,9035
***Text* $_{10}$**	0,8884	0,9030	0,8680	0,8785	0,9016	0,8871	0,8735	0,8963	0,9030	0,3341

TABLE 8: SUMMARY OF THE MAXIMUM DISTANCES OF TEXTS FROM ONE AUTHOR (ROGER MCHANEY)

NCD	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6
Row 1	0.8620	0.8628	0.8676	0.8690	0.8691	0.8722
Row 2	0.8724	0.8726	0.8726	0.8732	0.8737	0.8744
Row 3	0.8751	0.8768	0.8775	0.8778	0.8783	0.8787
Row 4	0.8799	0.8799	0.8799	0.8811	0.8819	0.8822
Row 5	0.8829	0.8877	0.8888	0.8900	0.8917	0.9035
Row 6	0.9035					

The compression method developed by *Chen, Li* and their colleagues is called the *Chen-Li Metric* (*CLM*) (Li et al., 2001;Chen et al., 2004), and the formulation is as follows:

$$CLM(x,y) = 1 - \frac{C(x) - C(x|y)}{C(xy)} \qquad (2.8)$$

Where x and y represents the objects that are to be compressed by algorithm C, and$C(x|y) = C(xy) - C(y)$. The result of this metric is between 0 and 1. When x and y are completely the same, $CLM(x,y)$ is 0, and when x and y are completely different, $CLM(x,y)$ is 1. Table 9 illustrates the *CLM* measure with 10 similar texts. Table 9 describes *CLM* measure with the same 10 texts that were used in the *CDM*.Among them, *Text*$_1$ and *Text*$_9$ were written by author Roger McHaney, *Text*$_6$ and *Text*$_7$ were written by author Ramaswamy Palaniappan, and *Text*$_3$ and *Text*$_4$ were written by author Weiji Wang. Compression distances from the same author are colored red. From the Table 9, it can be seem that the lowest values are from the diagonal line, which are the compression distances from the texts to themselves. The values in red are smaller than the values in black the same row or column. Table 10 illustrates the maximum intra-compression distances of the 31 texts from the same author (Roger McHaney) who wrote the $Text_1$ and $Text_9$. Therefore, in total 31 values sorted in an ascending order are shown in Table 10. Some values (e.g. 0.9429) in Table 10 are higher than a few values (e.g. 0.9075) in the Table 9, which indicates that *CLM* intra-compression distances sometimes are larger than *CLM* inter-compression distances.

TABLE 9: CLM MATRIX OF TEN TEXTS (COMPRESSION ALGORITHM: PPMD; BLUE REPRESENTS DIAGONAL VALUES; RED REPRESENTS DISTANCES FROM THE SAME AUTHOR)

CLM	$Text_1$	$Text_2$	$Text_3$	$Text_4$	$Text_5$	$Text_6$	$Text_7$	$Text_8$	$Text_9$	$Text_{10}$
$Text_1$	0,5435	0,9296	0,9407	0,9446	0,9346	0,9324	0,9355	0,9317	0,9233	0,9383
$Text_2$	0,9313	0,5358	0,9501	0,9582	0,9375	0,9436	0,9498	0,9107	0,9424	0,9512
$Text_3$	0,9391	0,9472	0,5039	0,8853	0,9495	0,9322	0,9316	0,9386	0,9506	0,9276
$Text_4$	0,9451	0,9563	0,8862	0,4845	0,9488	0,9362	0,9280	0,9465	0,9587	0,9346
$Text_5$	0,9390	0,9380	0,9518	0,9518	0,5544	0,9547	0,9503	0,9416	0,9481	0,9540
$Text_6$	0,9324	0,9416	0,9295	0,9354	0,9510	0,5056	0,9217	0,9351	0,9475	0,9386
$Text_7$	0,9308	0,9457	0,9299	0,9252	0,9436	0,9189	0,4942	0,9431	0,9544	0,9281
$Text_8$	0,9337	0,9075	0,9414	0,9503	0,9411	0,9356	0,9458	0,5268	0,9522	0,9475
$Text_9$	0,9225	0,9381	0,9493	0,9548	0,9432	0,9471	0,9549	0,9477	0,5061	0,9471
$Text_{10}$	0,9377	0,9469	0,9264	0,9327	0,9451	0,9373	0,9291	0,9441	0,9469	0,5009

TABLE 10: SUMMARY OF THE MAXIMUM DISTANCES OF TEXTS FROM ONE AUTHOR (ROGER MCHANEY)

CLM	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6
Row 1	0.9238	0.9259	0.9271	0.9276	0.9280	0.9286
Row 2	0.9286	0.9296	0.9296	0.9299	0.9302	0.9307
Row 3	0.9309	0.9310	0.9314	0.9316	0.9334	0.9334
Row 4	0.9334	0.9338	0.9345	0.9347	0.9352	0.9354
Row 5	0.9355	0.9356	0.9380	0.9383	0.9392	0.9429
Row 6	0.9429					

Another relatively new compression measure was developed based on cosine-vector dissimilarity measure by Sculley and Brodley (2006). The formulation of the *Compression-based Cosine* (*CosS*) metric is as follows:

$$CosS(x,y) = 1 - \frac{C(x) + C(y) - C(xy)}{\sqrt{C(x)C(y)}} \qquad (2.9)$$

The result of the *CosS* measure is in the range [0, 1], where 0 indicates a complete similarity and implies complete dissimilarity. Table 11illustrates the *CosS* measure by applying it to 10 texts. Table 11 describes *CosS* measure with the same 10 texts that were used in the *CDM*. $Text_1$ and $Text_9$ were written by author Roger McHaney, $Text_6$ and $Text_7$ were written by author Ramaswamy Palaniappan, and $Text_3$ and $Text_4$ were written by author Weiji Wang. Compression distances from the same author are colored red. From the Table 7, it can be seem that the lowest values are from the diagonal line, which are the compression distances from the texts to themselves. The values in red are smaller than the values in black (except the one on the diagonal line) from the same row or column.

Table 12 illustrates the maximum intra-compression distances of the 31 texts from the author Roger McHaney (one maximum value fore each text). Therefore, in total 31 values sorted in an ascending order are shown in Table 8. Some values (e.g. 0.8912) in Table 12 are higher than a few values (e.g. 0.8826) in the Table 11, which indicates that *CosS* intra-compression distances sometimes are larger than *CosS* inter-compression distances.

TABLE 11: COSS MATRIX OF 10 TEXTS (COMPRESSION ALGORITHM: PPMD; BLUE REPRESENTS DIAGONAL VALUES; RED REPRESENTS DISTANCES FROM THE SAME AUTHOR)

CosS	*Text* $_{1}$	*Text* $_{2}$	*Text* $_{3}$	*Text* $_{4}$	*Text* $_{5}$	*Text* $_{6}$	*Text* $_{7}$	*Text* $_{8}$	*Text* $_{9}$	*Text* $_{10}$
Text $_{1}$	0,3732	0,8684	0,8876	0,8946	0,8766	0,8733	0,8788	0,8720	0,8574	0,8835
Text $_{2}$	0,8715	0,3660	0,9047	0,9194	0,8817	0,8932	0,9044	0,8360	0,8910	0,9068
Text $_{3}$	0,8847	0,8994	0,3368	0,7941	0,9038	0,8726	0,8714	0,8842	0,9056	0,8648
Text $_{4}$	0,8955	0,9161	0,7957	0,3197	0,9027	0,8797	0,8651	0,8983	0,9204	0,8772
Text $_{5}$	0,8843	0,8827	0,9080	0,9079	0,3835	0,9129	0,9048	0,8893	0,9008	0,9120
Text $_{6}$	0,8733	0,8897	0,8678	0,8782	0,9060	0,3383	0,8549	0,8781	0,9003	0,8843
Text $_{7}$	0,8705	0,8970	0,8686	0,8604	0,8926	0,8500	0,3282	0,8922	0,9128	0,8657
Text $_{8}$	0,8756	0,8306	0,8891	0,9051	0,8883	0,8790	0,8972	0,3576	0,9087	0,9003
Text $_{9}$	0,8561	0,8835	0,9032	0,9132	0,8920	0,8994	0,9137	0,9006	0,3388	0,8995
Text $_{10}$	0,8826	0,8990	0,8628	0,8737	0,8957	0,8820	0,8675	0,8942	0,8990	0,3341

TABLE 12: SUMMARY OF THE MAXIMUM DISTANCES OF TEXTS FROM ONE AUTHOR (ROGER MCHANEY)

CosS	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6
Row 1	0.8583	0.8619	0.8641	0.8649	0.8657	0.8661
Row 2	0.8665	0.8683	0.8685	0.8689	0.8696	0.8700
Row 3	0.8707	0.8709	0.8714	0.8719	0.8750	0.8751
Row 4	0.8751	0.8758	0.8771	0.8774	0.8782	0.8785
Row 5	0.8786	0.8788	0.8831	0.8838	0.8851	0.8912
Row 6	0.8912					

CDM, *NCD*, *CLM* and *CosS* are four rather complex and different measures, however the actual difference mainly lies in the normalizing terms (Sculley & Brodley, 2006). The texts from the same author tend to have lower compression distances compared with the texts written by different authors. Nevertheless, this is not always the case. Hence, an effective model utilizing compression features should be able to separate texts written by the same author from texts written by other authors.

2.3.3 STYLOMETRIC FEATURES VS COMPRESSION FEATURES

Stylometric features are commonly used by linguistic experts to conduct authorship analysis. Moreover, stylometric features indeed reflect the writing styles of an author. Compared to the compression features, stylometric features are like a white box. In another word, they are more meaningful than compression features.

In terms of the efficiency, compression features cost less effort. In order to use stylometric features, first we have to decide what types of features we are going to use, which features of each type we are actually going to include in the model. What is equally important is the feature selection method, which is not a trivial task at all. On the contrary, compression features have much fewer constraints. Given two texts, compression distances can be immediately calculated using one of the compression distance measures explained in the previous part. All in all, it would improve the efficiency of an authorship attribution task.

2.4 COMPUTATION METHODS

The authorship analysis techniques include univariate, multivariate statistics, and machine learning techniques such as Support Vector Machine, Decision Trees, Neural Nets (Iqbal, Fung, Khan, & Debbabi, 2010). On essence machine learning techniques are multivariate statistics. The first computation method was a univariate approach, and the failure of which lead to the emergence of multivariate approach, and the latest machine learning techniques have facilitated the authorship attribution substantially (Koppel, Schler, & Argamon, 2009).

2.4.1 UNIVARIATE APPROACH

The scientific authorship analysis can be dated back to the late 19th century, and the main idea was that the authorship could be determined by the relationship of word length with the relatively occurrence frequencies (Koppel, Schler, & Argamon, 2009). In early 20th century, some statistic researchers tried to find the invariant properties in the written texts and this generated the idea for the authorship analysis that these invariant features might be used to solve authorship problems, whereas later using the invariant features to determine the authorship proved to be ineffective and gave way to the multivariate approach (Koppel, Schler, & Argamon, 2009).

2.4.2 MULTIVARIATE APPROACH

In 1964, Mosteller and Wallace stated new methods to solve the authorship attribution problem by combining multiple stylometric features, and this was believed to be the start of the multivariate approach (Koppel, Schler, & Argamon, 2009). It is said that Mosteller and Wallace used mainly the function words which were content-independent and applied Bayesian classification techniques to solve the authorship attribution problem, and the result was rather reliable (Koppel, Schler, & Argamon, 2009).

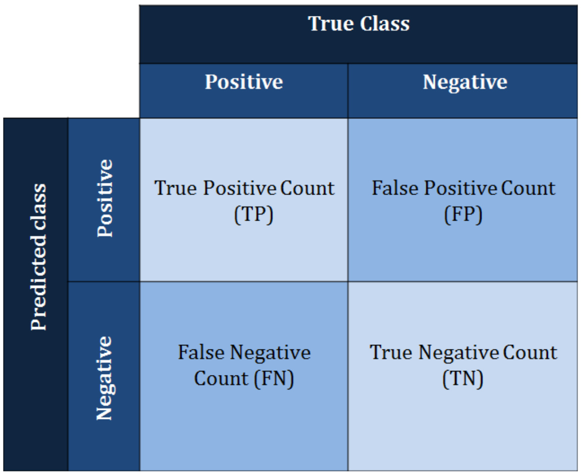
The advent of the machine learning techniques, especially the text-categorization techniques facilitated the authorship analysis (Koppel, Schler, & Argamon, 2009). The idea of applying text categorization techniques to solve authorship analysis problem is that transforming the training dataset into feature vectors and using the text categorization techniques to set the boundaries of the target class and the outliers (Koppel, Schler, & Argamon, 2009). Texts can be regarded as vectors in multi-dimensional space. Thus statistical and machine learning techniques such as *Discriminant Analysis*, *Support Vector Machines*, *Decision Tress*, *Neural Networks*, *Generic Algorithms*, *memory-based learners*, *classifier ensemble methods* can be employed to train the classification models (Stamatatos, 2009).

According to the extensive literature review conducted by Koppel et al. (2009), support vector machine has been proved at least as good as the other machine learning techniques to solve authorship attribution problem, and meanwhile some Winnow and Bayesian regression techniques have been shown to have high potential to tackle the problem.

2.5 Performance Measures

The coincidence matrix (Figure 15) is the original source of the performance measures in the classification problem (Olson & Delen, 2008). The commonly used performance measures are *True Positive Rate*, *True Negative Rate*, *Accuracy, Precision*, *Recall* and *F-measure*, and the formulations of each are shown below (Olson & Delen, 2008). In Figure 15, *True Positive Count* and *True Negative Count* are the correct decisions made, while *False Positive Count* and *False Negative Count* are the wrong decisions.

The True Positive Rate is the True Positive Count divided by the sum of True Positive Count and False Negative Count; the True Negative Rate is the ratio of True Negative Count divided by the sum of True Negative Count and False Positive Count.



		True Class	
		Positive	Negative
Predicted class	Positive	True Positive Count (TP)	False Positive Count (FP)
	Negative	False Negative Count (FN)	True Negative Count (TN)

Figure 15: A coincidence matrix of the performance measures (Olson & Delen, 2008)

Accuracy is the ratio of the correct decisions divided by all the decisions, which is shown in Table 13. Another two performance measures: *recall* and *precision*, are widely used in information retrieval and other relevant fields. Precision indicates that to what extent the model can retrieve more relevant information than irrelevant information, while recall reflects the degree that relevant information is obtained. Thus, the precision is calculated by dividing True Positive Count (relevant information obtained) by the sum of True Positive Count and False Positive Count (irrelevant information obtained). Similarly, the recall is measured by dividing True Positive Count with the sum of True Positive Count and False Negative Count (relevant information missed). Since the improvement on recall can be traded-off by lowering precision, therefore F-measure is used to evaluate a model by balancing precision and recall.

How do the precision and recall trade off?

Imagine there is a competition of information retrieval, and the rank is determined based on the precision of the model designed. The strategy to improve the precision is to retrieve information only when it is very certain (e.g. 99% probability) that the information is relevant. What is the consequence of this strategy? Much relevant information will be missing since the model is not very sure about it. Consequently, the recall will be very low. However, this will not affect the precision of the model. Obviously, this is not the most desirable model designed to solve the real problem.

On the other hand, if the rank is determined according to the recall of the model designed, thus the strategy is to retrieve all the information. In this case, definitely all the relevant information will be retrieved, and nicely the model will get the best recall (value 1), whereas the precision is low. Again this is apparently not the most desirable model. Therefore, a performance measure F is created to balance these two performance measures. The harmonic average of the precision and the recall is called F_1, which is commonly used.

TABLE 13: PERFORMANCE MEASURES

Performance Measure	Formulation
True Positive Rate	$\frac{TP}{TP+FN}$
True Negative Rate	$\frac{TN}{TN+FP}$
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F-measure (F_1)	$\frac{2}{\frac{1}{Precision}+\frac{1}{Recall}}$

CHAPTER 3 MODEL DESIGN

3.1 AUTHOR VERIFICATION DESIGN

This authorship verification research is designed to use both one-class classification approach and two-class classification approach to solve the authorship verification task. As has been explained in the previous chapter, the main difference between the one-class classification approach and the two-class classification approach lies in the outlier class representation. The one-class classification approach does not require an outlier class representation, while for the two-class classification approach it is of significant importance, which means collecting relevant data to represent the outlier class is a critical step.

With regard to the one-class classification, a Parameter-based Model is designed, which takes the known texts as input and predicts the label ('Y', 'N') of the unknown text in each problem based on the statistic parameter μ (mean) and σ (standard deviation). Another one-class classification approach Distribution-based Model compares the similarity of two distributions and predicts the label ('Y','N') of the unknown text according to the result of the hypothesis testing. On the other hand, under the two-class classification approach, two different models are designed: Instance-based Compression Model and Character N-Gram Model. Three models are included in the Instance-based Compression Model, i.e. Naïve Approach, Compression Feature prototype Model and Bootstrapping Approach.

In terms of the compression distance measures, only those that are explained in paragraph 2.3.2 are adopted. With the aspect to the stylometric features, only character *n*-grams (*n*=2, 3), which are known for their outstanding performance (Grieve, 2007), are experimented in this research. The overview of the framework is shown in the Figure 16.

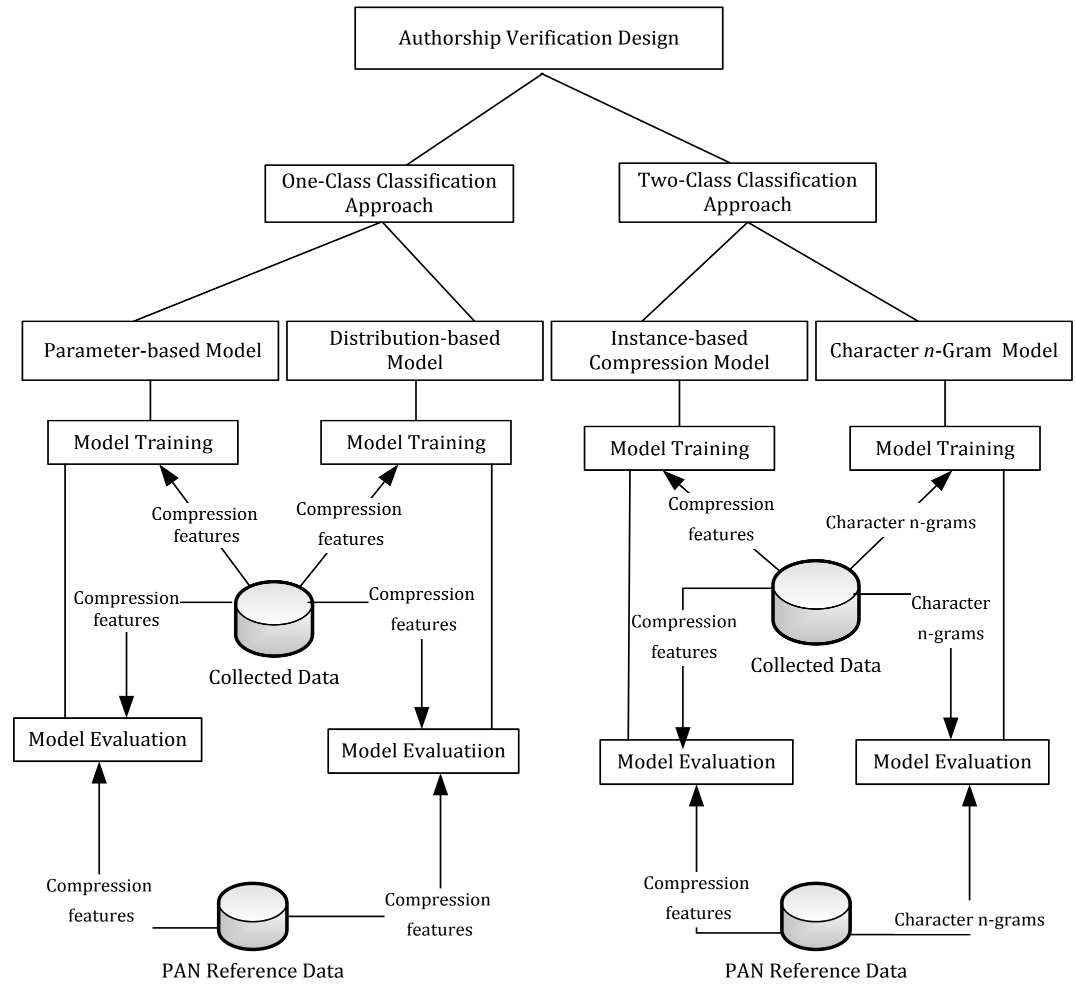


FIGURE 16: OVERVIEW OF THE RESEARCH DESIGN

With regard to the compression algorithms, a variant of the *Prediction by Partial Matching* (*PPM*) called *PPMd* is selected. *PPM*, which is developed by CLEARY & WITTEN (1984), is among the most promising lossless data compression alogrithms (Shkarin, 2002). *PPM*outperforms other compression algorithms in terms of English text compression (MOFFAT, 1990). Regarding to the Character N-Gram Model, character bigrams and trigrams are selected.

3.2 ONE-CLASS CLASSIFICATION APPROACH DESIGN

The idea of one-class classification is to make the most use of the given known texts and set a classification rule to determine the label of the unknown text. In this research, two models are designed as a one-class classification method utilizing compression distance. For this model, the following four compression distance metrics from the Table 14 are selected.

TABLE 14: COMPRESSION DISTANCE METRICS

Name	Equation
Normalized Compression Distance	$NCD(x,y) = \frac{C(x+y) - \min\{C(x),C(y)\}}{\max\{C(x),C(y)\}}$
Chen-Li Metric	$CLM(x,y) = 1 - \frac{C(x) - C(x\|y)}{C(xy)}$
Compression-based Cosine	$CosS(x,y) = 1 - \frac{C(x) + C(y) - C(xy)}{\sqrt{C(x)C(y)}}$
Compression-based Dissimilarity Method	$CDM(x,y) = \frac{C(xy)}{C(x) + C(y)}$

3.2.1 PARAMETER-BASED MODEL

The original idea of this method is that if several texts are drafted by the same author, then even a *small part of a text* can reflect the author's specific writing style (specific to the contents or topics etc.), while the entire text which can be attained by concatenating the rest of documents can well represent the author's general writing style (the effects of a specific topic or content can be smoothed out when there are various topics or contents). The compression distance from the small text to the entire text is regarded as the interior distance. As different parts of the documents are selected as the small document, the range of the interior distance improves.

The rationale is:

Within the given limited data, an interior distance can be defined and if the distance of the unknown text to the entire document (average is used since the entire document keeps changing slightly) falls in the interior distance range, then the author of the known documents is also the author of the unknown document.

The original texts from the target class are divided into q pieces. This step is designed in case there is only one known document, which is a situation the rest of the processes cannot go on. Therefore if one problem has n ($1 \le n \le 10$) known texts, after this treatment qn texts are created, each time one text from the target class is selected and the rest (qn-1) texts are concatenated into one text T. After that, compression distances ***from*** both the selected text and the unknown text ***to*** the concatenated text T are computed. This process is iterated till all the qn texts derived from the target class are selected once. After the iteration is completed, two compression distance vectors are generated. The compression distance vector of the texts from the target class is denoted with M= [d_1, d_2, d_3...d_{qn}], which is a qn-dimensional vector; the compression distance vector of the unknown texts to the concatenated text T is denoted by M'= [d'_1, d'_2, d'_3...d'_{qn}] that includes all the compression distances from each iteration.

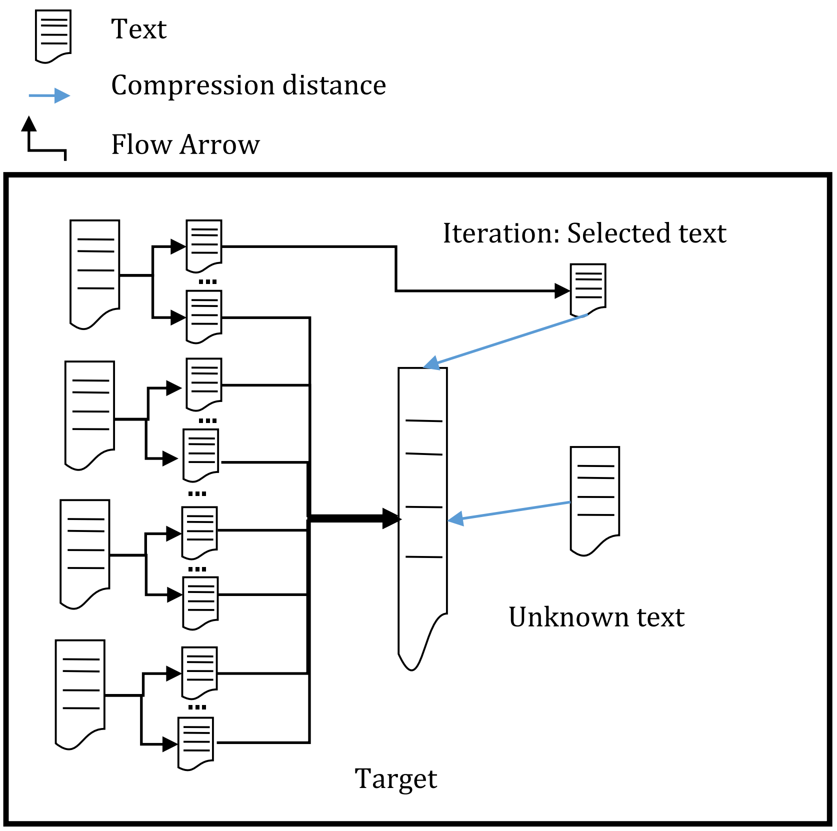


FIGURE 17: ONE ITERATION ROUND OF THE ONE-CLASS CLASSIFICATION COMPRESSION APPROACH

In Figure 18 and Figure 19, the distribution on the left is from *M*, and the distribution on the right is from *M*'. The standard deviation σ for *M'* is much smaller (e.g. 100 times smaller) than that of *M*. This is because the compression distances in *M'* are always from the same unknown text to the slightly changed concatenated *T*. Nevertheless, the compression distances in *M* are from the selected text to the newly concatenated text *T*, both of which keeps changing. Consequently, the threshold of the classification is as follows:

$$\mu' <= \mu + a\sigma$$

Where μ' is the mean of the matrix *M'*, μ is the mean of *M*, σ is the standard deviation of *M*, and *a* (*a*>=1) is the factor of σ. In a normal distribution 95% of the values fall within two standard deviation of the mean (68% of the values fall within one standard deviation of the mean). Hence if the μ' lies out of the two standard deviations of *M*, there is only 5% probability that it belongs to *M*, and this is the acceptable error rate (false negative) that this research would accept. Nonetheless, this might pose a potential risk of relatively high false positive rates. To make a comparison of different values of *a*, two values of *a* (*a*=1; *a*=2) are selected in this model.

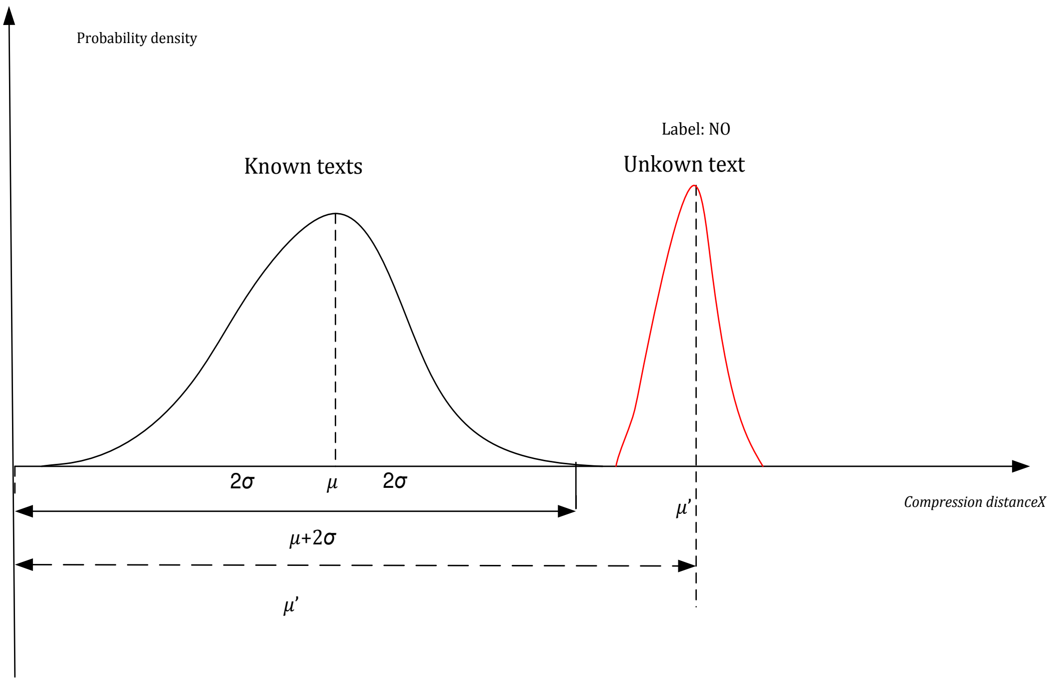


FIGURE 18: DENSITY DISTRIBUTIONS WHEN THE UNKNOWN CAN BE LABELED WITH NO

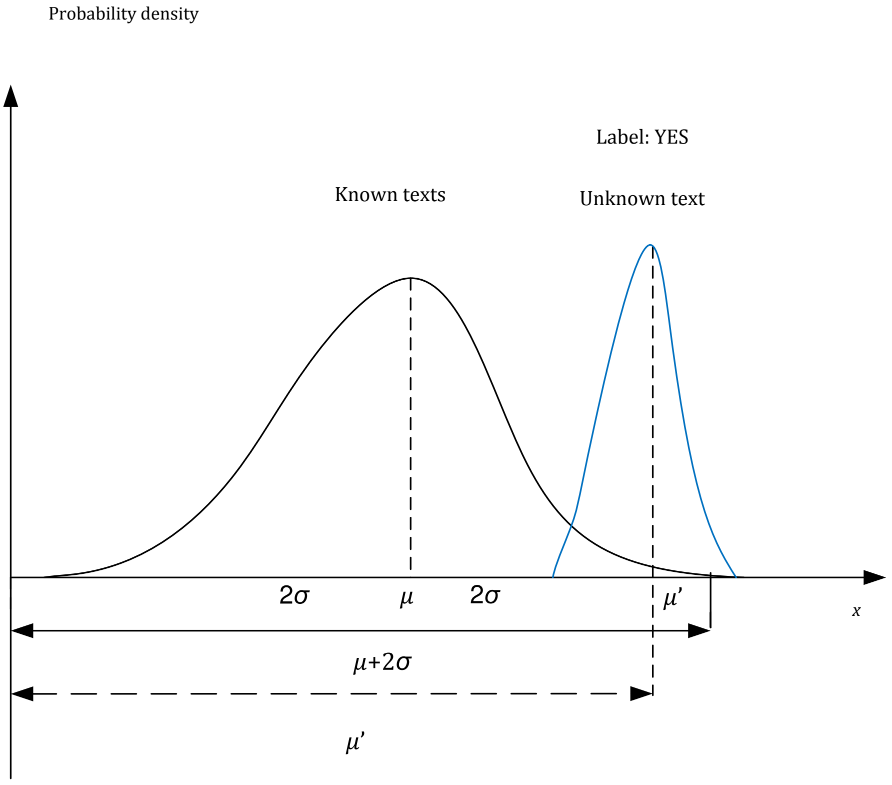


FIGURE 19: DENSITY DISTRIBUTIONS WHEN THE UNKNOWN CAN BE LABELED WITH YES

3.2.2 DISTRIBUTION-BASED MODEL

Different from the Parameter-based Model, the Distribution-based Model does not have the text concatenation procedure. Instead, all the compression distances between two texts are calculated. For instance, there are $qn\in[q, 10q]$ split known texts ($n\in[1, 10]$, texts are split into q pieces), consequently $qn(qn\text{-}1)/2$ compression distances are computed. Theoretically, the order of two texts in a compression measures does not affect the result. Nonetheless, in practice the results of compression distance from text x to text y and from text y to text x are slightly different. As is shown in Figure 20, this trivial difference is ignored in this design when computing the compression distance between two known texts. After completing the compression distance computation, two vectors are generated. Vector V_t consists of all the compression distances between two known texts, and the other vector V_o consists of the compression distances from the unknown text to all the known texts.

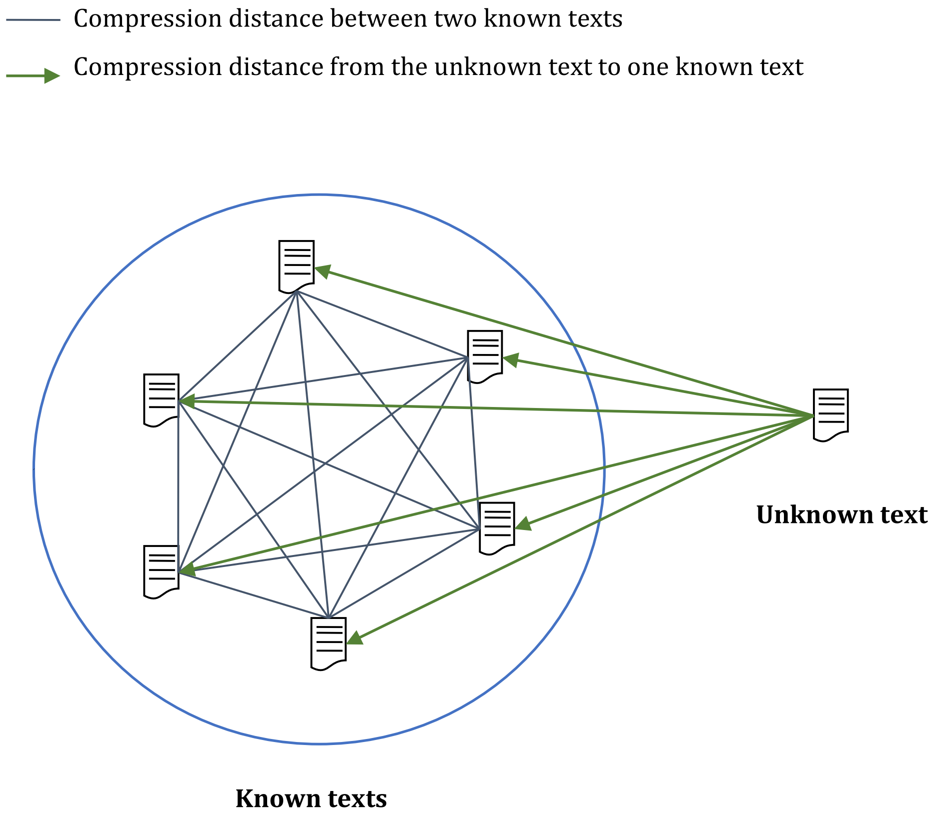

Compression distance between two known texts
Compression distance from the unknown text to one known text
Unknown text
Known texts

FIGURE 20: INSTANCE-BASED CLASSIFICATION MODEL

THE RATIONALE BEHIND THIS IS:

If the vector V and V' follow the same continuous distribution, then the unknown text is written by the author who wrote the known texts.

TWO-SAMPLE KOLMOGOROV-SMIRNOV TEST

Two-sample Kolmogorov-Smirnov test (Massey, 1951; Miller, 1956) is selected to test whether the vector V and V' are from the same continuous distribution.

$$D\ n,\ n' = \text{maximum}\ \left|F_{1,n}(x) - F_{2,n'}(x)\right|$$

Where $F_{1,n}(x)$ is the empirical distribution function of the first data sample, and $F_{2,n'}(x)$ is the empirical distribution function of the second data sample. The decision to accept or reject is referred to a three-dimensional table (n, n', α (the probability of type I error, i.e. rejecting the null hypothesis when it is true)) (Massey, 1951).

3.3 TWO-CLASS CLASSIFICATION APPROACH DESIGN

One type of the two-class classification models makes use of compression features, while one separate model is designed with character n-grams. As has been mentioned before, to solve the research problem as a two-class classification problem the data collection and preparation are of great importance.

3.3.1 INSTANCE-BASED COMPRESSION MODEL

For the task of computing the compression distance, the compression-based dissimilarity method, developed by Keogh et al. (2004) is adopted as the major compression distance measure:

$$CDM(x, y) = \frac{C(xy)}{C(x) + C(y)}$$

Another compression distance measure *CosS* is also implemented to make a comparison. Three different models have been designed: Naïve Approach, Compression Feature Prototype Model, and Bootstrapping Approach.

NAÏVE APPROACH

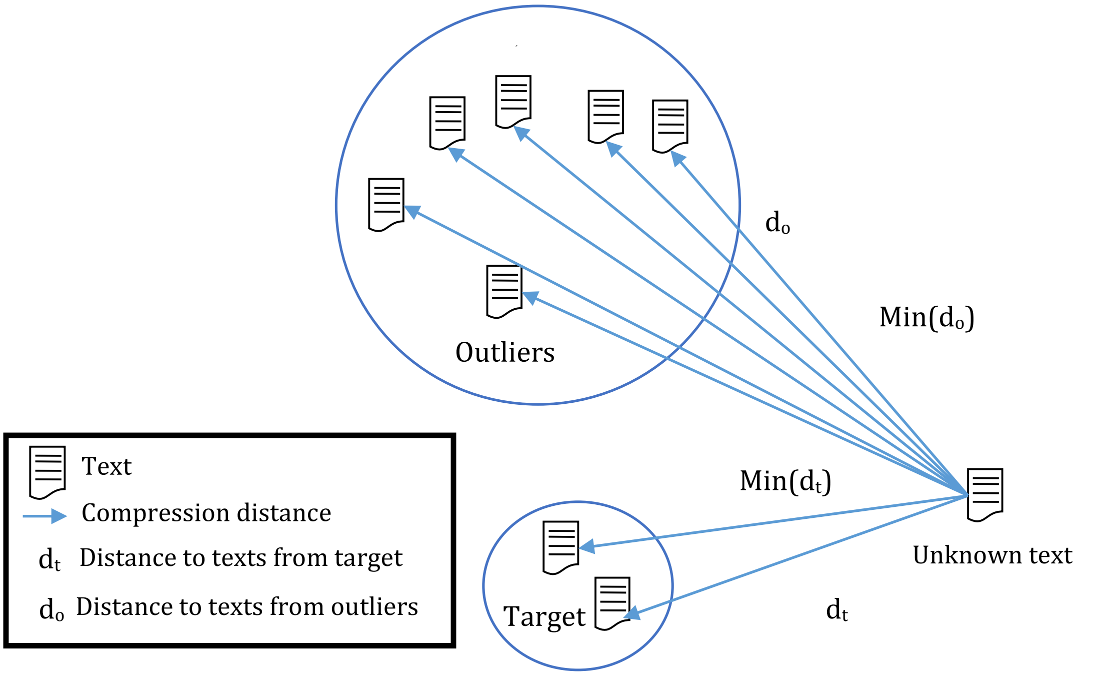


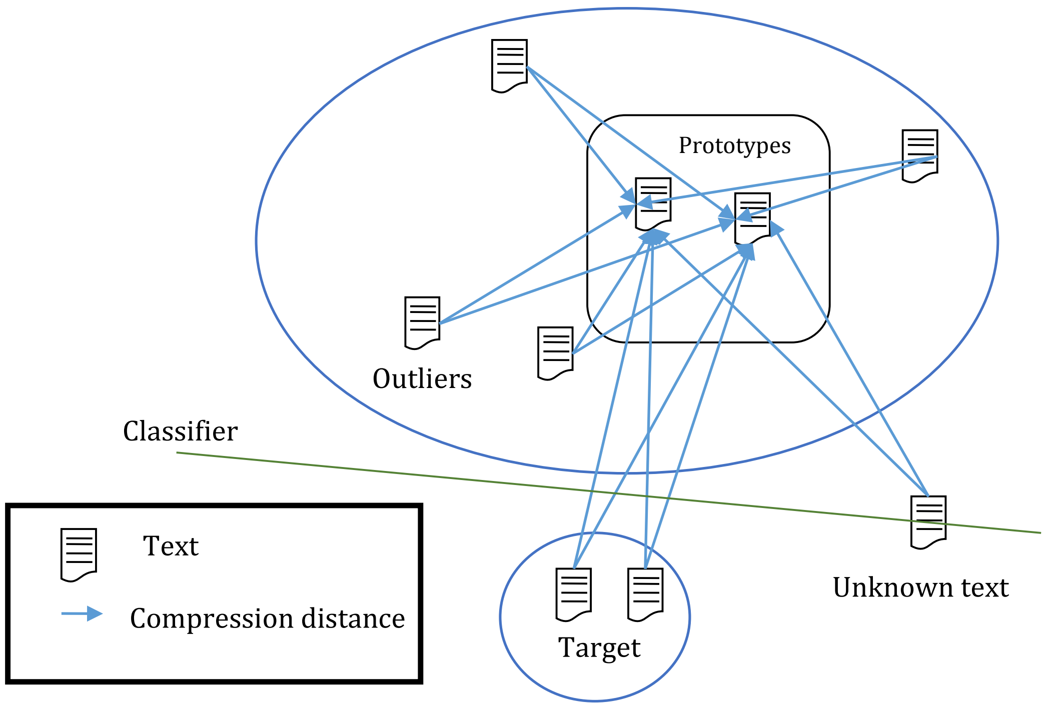
FIGURE 21: K-NEAREST NEIGHBOR MODEL

Figure 21 describes the naïve compression approach. The compression distances of the unknown text to all the texts from the outlier class and target class are measured based on the compression measures described before. All the texts from the outlier class and the target class are variable y in the equation, and the single unknown text is variable x in the compression measure formulation. Whether the unknown text belongs to target or outlier class depends on the minimum compression distance to the texts from the outlier class (d_o) and to the texts from the target class (d_t). If min (d_o) is smaller than min (d_t), then it means that the unknown text is closer to the outlier class and thus it belongs to the outlier; if min (d_o) is larger than min (d_t), therefore the unknown text is more similar to the target class and thus it will be labeled with Y, which means the unknown text is indeed written by the same author who has written all the texts in the target class.

The rationale of this model:

Similar to the k-nearest neighbor algorithm (k=1 in this model), the closeness of the unknown text to the target class and outlier class is measured by compression distance, and the unknown text belongs to whichever has the smaller compression distance.

COMPRESSION FEATURE PROTOTYPE MODEL



Note: The number of text documents from the outlier class and the target class are not the exact numbers. A few samples are selected to visualize the process.

FIGURE 22: COMPRESSION FEATURE PROTOTYPES APPROACH

A set of the data, which is called prototypes, is sufficient to represent the whole data (Duin, et al., 2007). The compression features are also regarded as lexical level stylometric features (Lambers & Veenman, 2009). Therefore, some prototypes are selected from the outlier class. The number of prototypes that should be selected is a critical parameter that needs to be tuned. The basic idea of this approach is separating the target class from the outlier class according to the compression distance to the prototypes.

The rationale of this model is:

With regard to the compression distance from target class and outlier class to prototypes (separately), there is a significant difference between target class and outlier class, and based on the compression distance from the unknown text to prototypes, the unknown text can be correctly labeled with either target class or outlier class.

As is shown in Figure 22, the compression distances D from all the texts T in the *outlier class*, in the *target class* and the single *unknown text* to prototypes are computed with *CDM*. The result for each text is a vector containing the compression distance to each prototype. Furthermore, when all the compression distance vectors V from the same class are concatenated together, a compression distance matrix M for each class is generated. The compression distance matrix of the target class is annotated with M_t, and the compression distance matrix of the outlier class is annotated with M_o. After that, M_o and M_t are concatenated to reach a training dataset *Train_Dataset*, with all the vectors from M_o

labeled 0 and all the vectors from the M_t labeled 1. The training dataset is trained with a machine learning classifier to find the boundary between these two classes, and eventually the compression distance vector of the unknown text is applied the trained mapping, and the label of the unknown text can be decided. Some mathematical explanation is as follows:

Compression distance vector $v= [d_1, d_2...d_n]$, is a vector of compression distances of one text to all the prototypes and hence n equals to the number of prototypes.

$$M_{\mathrm{t}}=\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \cdot \\ . \\ . \\ v_q \end{bmatrix} M_0=\begin{bmatrix} v_1' \\ v_2' \\ v_3' \\ . \\ . \\ . \\ v_p' \end{bmatrix}$$

FIGURE 23: THE COMPRESSION DISTANCE MATRIX OF TARGET CLASS (LEFT) AND THE COMPRESSION DISTANCE MATRIX OF OUTLIER CLASS (RIGHT)

M_{t} is the compression distance matrix of the target class containing all the compression distance (from the target class to the prototypes) vectors (from v_1 to v_q), and q equals to the number of texts in the target class; M_0 is the compression distance matrix formed by all the compression distance (from outliers to the prototypes) vectors (from v_1' to v_p') of the outlier class, and p equals to the number of texts in the outlier class. The variables in the *Train_Dataset* are as follows:

LABEL	VALUE
$\begin{bmatrix} 1 \\ 1 \\ 1 \\ . \\ . \\ . \\ 1 \end{bmatrix}$	$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ . \\ . \\ . \\ v_q \end{bmatrix}$
$\begin{bmatrix} 0 \\ 0 \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix}$	$\begin{bmatrix} v_1' \\ v_2' \\ v_3' \\ . \\ . \\ . \\ v_p' \end{bmatrix}$

FIGURE 24: THE TRAIN_DATASET

When the *Train_Dataset* is well prepared, a linear classifier *LESS* (Lowest Error in a Sparse Subspace) developed by Veenman and Tax (2005) is applied to the *Train_Dataset* to obtain the trained mapping. It is said that *LESS* classifier can efficiently find the linear discriminants in a sparse subspace (Veenman & Tax, 2005). *LESS* classifier is regarded as a weighted Nearest Mean Classifier, and it can also be seen as a variant of the L_1 Support Vector Machine. The details of *LESS* classifier will be explained later.

BOOTSTRAPPING APPROACH

The Bootstrapping Approach can be seen as a variant of the Compression Feature Prototype Model. One critical constraint of the research problem is that the known texts for each author are very limited (one to ten known documents for each author). Therefore, by merging the known text documents from one author and resampling them with different size, many more texts documents can be generated in this way. This is the original idea of the bootstrapping approach.

The rationale of this approach is:

The number of known documents has impact on the accuracy of the modeling, since each text is treated as an individual object. Thus, by generating more known documents can improve the performance of the model.

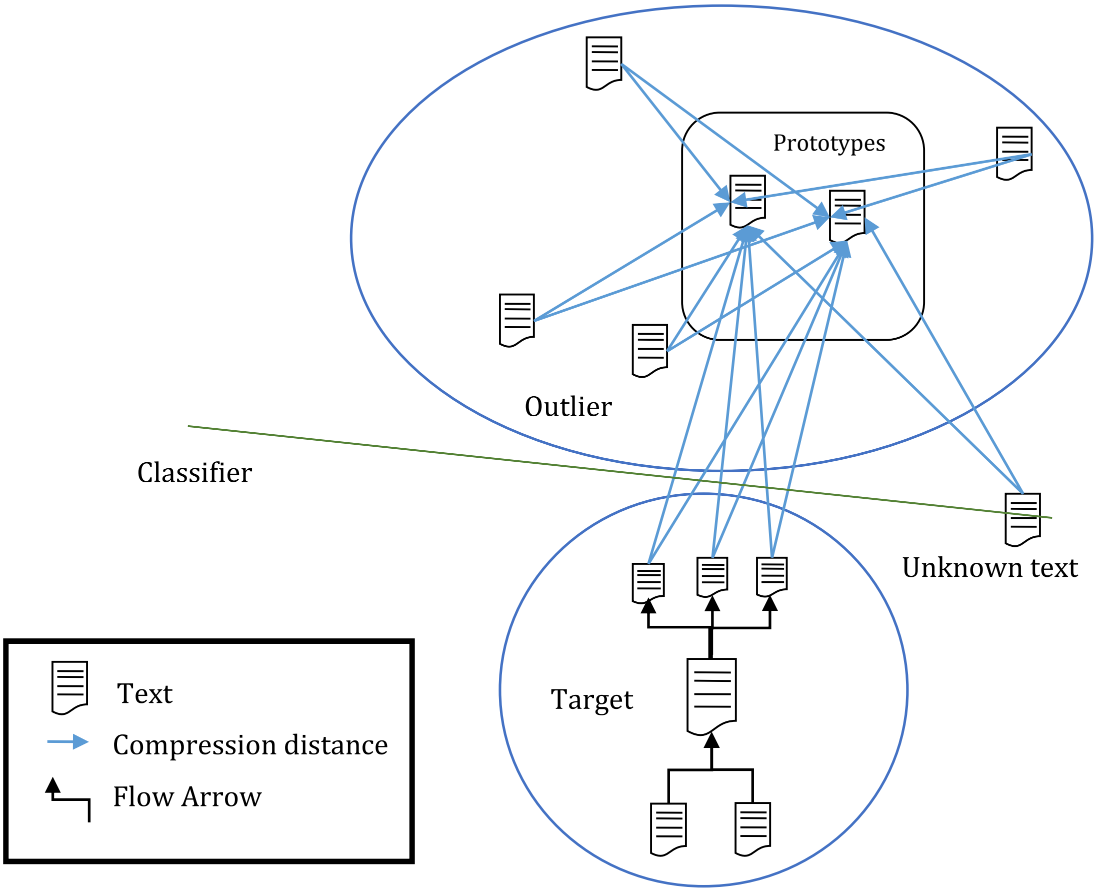


FIGURE 25: BOOTSTRAPPING APPROACH

Note: The number of text documents from the outlier class and the target class are not the exact numbers. A few samples are selected to visualize the process.

When using prototypes, the result might be biased by selecting some prototypes, therefore to reduce the effects of these prototypes, 10 rounds of iteration is planned to the Compression Feature Prototype Model as well as the Bootstrapping Approach.

LESS CLASSIFIER

LESS classifier developed by Veenman and Tax (2005), is a variant of the linear support vector machine. The formulation is as follows (Veenman & Li, 2013):

$$min \sum_{j=1}^{p} w_j + C(\sum_{i=1}^{n_t} \xi_{ti} + \sum_{i=1}^{n_o} \xi_{oi})$$

$$\text{Subject to:} \quad \begin{cases} x \ \in X_t, \ \sum_{j=1}^{p} w_j f(x,j) \geq 1 - \xi_{ti} \\ x \in X_o, \sum_{j=1}^{p} w_j f(x,j) < -1 + \xi_{oi} \end{cases}$$

Where$f(x,j)$=$(x_j - \mu_{tj})^2 - (x_j - \mu_{oj})^2$, w_j≥0, $\xi_{ti} \geq 0$,, $\xi_{oi} \geq 0$.

w_j is the weight of the dimension j (j∈ [1, p]), p is the number of dimensions, X_t is the target class with n_t elements, X_o is the outlier class with $n_o$elements, μ_{tj} is the mean of the target class at the dimension j, and μ_{oj} is the mean of the outlier class at the dimension j. ξ_{ti} and ξ_{oi} are slack variables that allow error to some extent, and C is a tunable regularization parameter.

3.2.2 CHARACTER N-GRAM MODEL DESIGN

Character *n*-grams (*n*=2, 3) are the selected stylometric features. The 26 lowercase alphabets and 26 uppercase together with the whitespace are included in the design of the character *n*-gram model. Additionally, some basic punctuation marks (( ) . , : ; - ? ') are also added to the study.

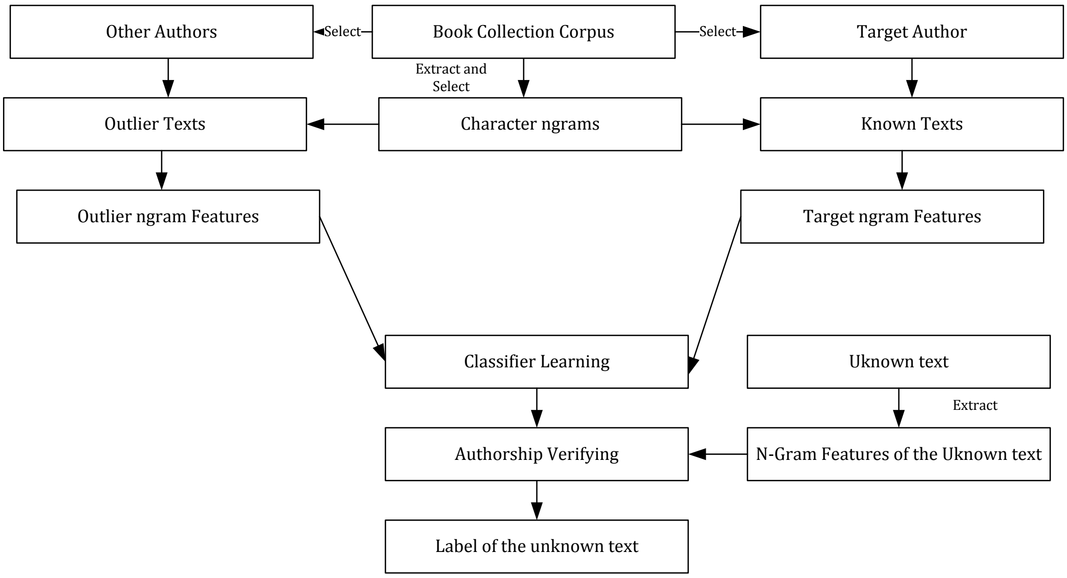


FIGURE 26: OVERVIEW OF THE DESIGN OF THE CHARACTER N-GRAM MODEL

The set of the character *n*-grams of the entire Book Collection Corpus excluding the target author is denoted with C= $\{c_o\}$, with one element representing the character *n*-grams of the entire corpus. The set of the character *n*-grams of the target author is denoted with C_t = $\{c^t_1, c^t_2... c^t_m\}$, where c^t represents the character *n*-grams of the known texts from the target class, and m is the number of known texts in the given problem.

The *n*-gram features are selected based on their relative frequencies (e.g. character trigram '*ded*' occurs 2271 in the bag of character *n*-grams $\mathrm{c_o}$, which in total has 6,948,979 character trigrams (including repetition trigrams), and thus the relative frequency of trigram '*ded*' is 0.0327%, and consequently the sum of the relative frequencies of the unique trigrams is 1).

The feature vectors have both labels of the features and their corresponding values. In the case of character *n*-grams, the feature labels are the selected character *n*-grams, and the values are their corresponding relative frequencies. The labels of character *n*-gram features are denoted with a vector $\boldsymbol{l}$= $[l_1, l_2 ... l_p]$, and their values are denoted with a vector $\boldsymbol{f}$= $[f_1, f_2 ... f_p]$. Consequently, the feature vector of the outlier class is represented with v'= $[f'_1, f'_2 ... f'_p]$, and the value vector of one known text document from the target class is denoted with $\boldsymbol{f_q}$= $[f^k_{q1}, f^k_{q2} ... f^k_{qp}]$, where p is the number of known texts. The frequency value vector of the unknown text is $\boldsymbol{f_u}$= $[f^u_1, f^u_2 ... f^u_p]$.

As has been explained above, the character *n*-gram features are selected from the Book Collection Corpus with a threshold of their relative frequencies. After this step, a label vector $\boldsymbol{l}$= $[l_1, l_2 ...l_n]$ is selected. Character *n*-grams from the vector $\boldsymbol{l}$ are searched through the set of character *n*-grams of the target class C_t= {$c^k{}_{1,}$ $c^k{}_2$... $c^k{}_m$}. If the character *n*-gram l_i (i=1 to n) from the vector ***l*** can be found in $c^k{}_r$ (r=1 to m) or in c^u, then the relative frequency of that specific character *n*-gram will be retrieved, otherwise the frequency will be attributed with 0. After the search phase has completed, for the known texts from the target class, a frequency value matrix containing several frequency value vectors is generated, which is described in Figure 27 where *m* represents the number of the known texts and *q* is a number between 1 and *m*. When *m* is 1, thus the matrix becomes one vector. Besides, a frequency value vector $\boldsymbol{f^u} = \lfloor f_1^u,\ f_2^u,\ \dots\ f_n^u \rfloor$ for the single unknown text document is created.

$$\begin{bmatrix} f_{11}^k, f_{12}^k \cdots , f_{1p}^k \\ \cdot \\ \cdot \\ \cdot \\ f_{q1}^k, f_{q2}^k \cdots , f_{qp}^k \\ \cdot \\ \cdot \\ \cdot \\ f_{m1}^k, f_{m2}^k \cdots , f_{mp}^k \end{bmatrix}$$

FIGURE 27: A FREQUENCY VALUE MATRIX

In order to build a trained classifier, a *Train_Dataset* is prepared. A classifier is applied to the *Train_Dataset* to learn the boundary between the two classes to predict the class label of the unknown text based on its frequency value vector.

Class Label	Value
$\begin{bmatrix} 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}$	$\begin{bmatrix} f_{11}^k, f_{12}^k \cdots , f_{1p}^k \\ \cdot \\ \cdot \\ \cdot \\ f_{q1}^k, f_{q2}^k \cdots , f_{qp}^k \\ \cdot \\ \cdot \\ \cdot \\ f_{m1}^k, f_{m2}^k \cdots , f_{mp}^k \end{bmatrix}$
$\begin{bmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$	$\begin{bmatrix} f_{11}^k, f_{12}^k \cdots , f_{1p}^k \\ \cdot \\ \cdot \\ \cdot \\ f_{n1}^k, f_{n2}^k \cdots , f_{np}^k \end{bmatrix}$

FIGURE 28: TRAIN_DATASET OF THE CHARACTER N-GRAM MODEL

CHAPTER 4 EXPERIMENTATION

4.1 DATA COLLECTION AND PRE-PROCESSING

In order to have the outlier class, the data collection is essential. The collected data should have similar genre, date of writing and themes. Moreover, because the training dataset given by the PAN Contest is not large enough to build a reliable, the data collection is indispensable for the one-class classification approach as well. Based on a quick web search, most of the English texts PAN selected in the training corpus could be found on bookboon.com, which provides open access for students to text books (Kati, 2012). Consequently, the technical books provided on bookboon.com became the good candidates for the outlier class. All the textbooks from the Engineering (Chemical Engineering, Construction Engineering etc.) and IT & Programming were collected in the first step. After checking the collected book one by one, a few books were removed and eventually 72 books written by 51 authors were selected, five of which were authors of the PAN's training corpus. The full lists of books and authors can be found in the Appendix B. Figure 29 describes the steps that were taken in the data collecting and pre-processing phase.

First of all, the books were converted to plain text files. Whether the encoding of the text files would make a difference is uncertain, whereas it is safe to encode the plain text files to UTF8, which is in alignment with PAN's text files. After the plain texts files were encoded by UTF8, some text cleansing procedures such as removing all the advertisements which sporadically presented in the books and removing numbers which are said to provide little stylometric information. Moreover, book titles, chapter titles were removed semi-manually. The final step of the data preparation is text splitting. Till the step 3, each book was represented by one large text document, while the collected documents should be prepared in the same way as the training data provided by PAN Contest, and thus all the large text documents were split into small documents with a length between 6,000 characters and 8,000 characters, around 1,000 words each text document, resulting in 2 to 75 small text documents for each book.

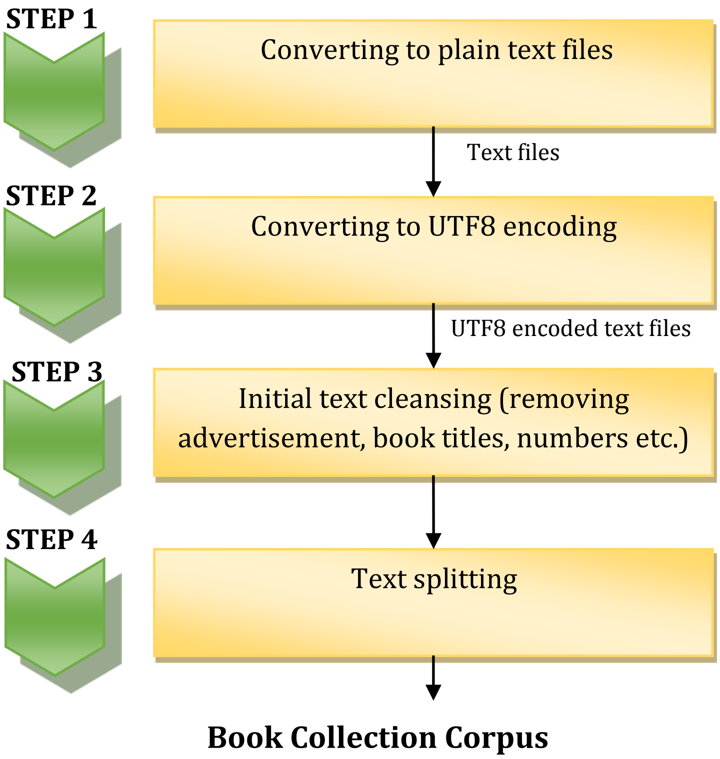
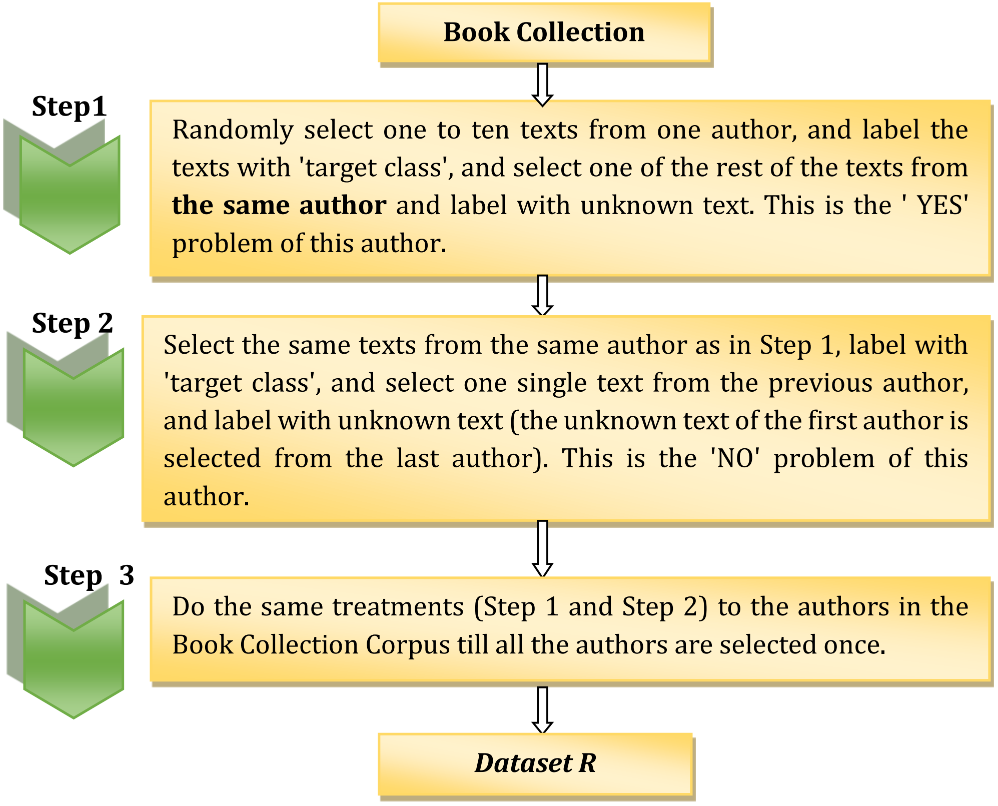

STEP 1
Converting to plain text files
Text files
STEP 2
Converting to UTF8 encoding
UTF8 encoded text files
STEP 3
Initial text cleansing (removing advertisement, book titles, numbers etc.)
STEP 4
Text splitting
Book Collection Corpus

FIGURE 29: DATA COLLECTION AND PRE-PROCESSING STEPS

The Dataset R is prepared to test the performance of the models designed with compression features. The preparation of the Dataset R is described in Figure 30. Since the selection of the unknown text for each problem is random, thus generating Dataset R several times result in different datasets. Within each Dataset R, there are 100 problems to label.

4.2 DESCRIPTIVE ANALYSIS



Book Collection

Step1

Randomly select one to ten texts from one author, and label the texts with 'target class', and select one of the rest of the texts from **the same author** and label with unknown text. This is the ' YES' problem of this author.

Step 2

Select the same texts from the same author as in Step 1, label with 'target class', and select one single text from the previous author, and label with unknown text (the unknown text of the first author is selected from the last author). This is the 'NO' problem of this author.

Step 3

Do the same treatments (Step 1 and Step 2) to the authors in the Book Collection Corpus till all the authors are selected once.

Dataset R

FIGURE 30: THE PREPARATION OF DATASET *R*

As explained in the previous chapter, the profile-based approach and the instance-based approach are two different data sampling approaches. The Parameter-based Model is a variant of the profile-based approach, which selects one known text and concatenates the rest, and computes the compression distance from the selected known text and the unknown text to the concatenated text. On the other hand, Distribution-based Model is an instance-based approach, which computes the compression distances among the known texts, as well as the compression distances from the unknown text to all the known texts.

In order to gain insights into the Book Collection Corpus and make a comparison of the two different models, a descriptive analysis is conducted as Figure 31 and Figure 32. The compression metric adopted is *CDM*. The results of conducting the descriptive analysis on all the 50 authors has shown that the profile-based data sampling approach reduces the overlap of the intra-compression distances with inter-compression distances. The overlap of the intra distances with the inter distances is difficult area for the designed models to set a rule to classify. Therefore, the smaller the overlapped area, the better the data sampling method is. An example of comparison is shown in Figure 33. More comparison results can be found in the Appendix C.

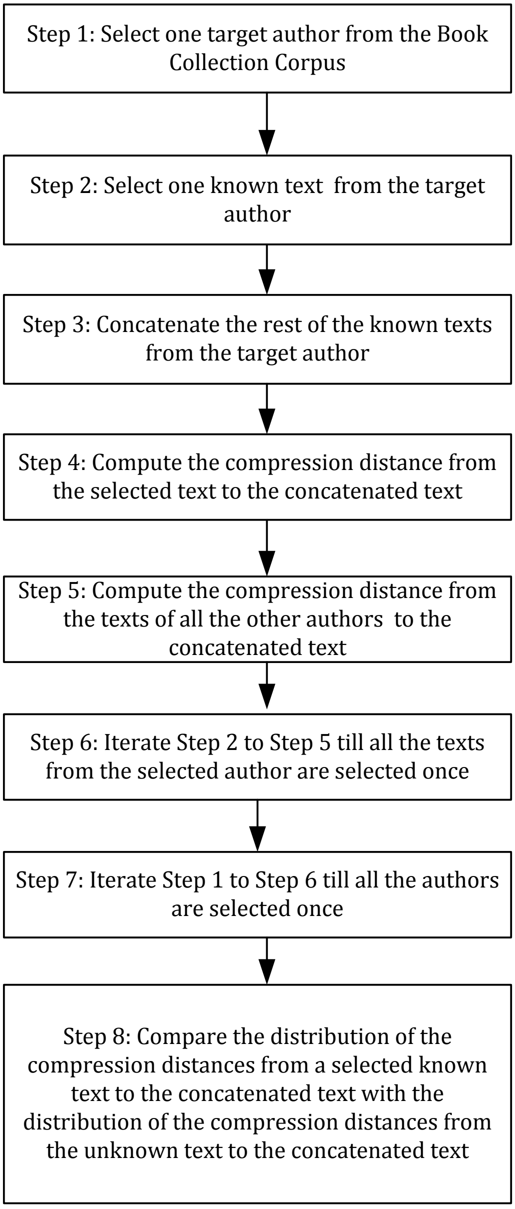

Step 1: Select one target author from the Book Collection Corpus
Step 2: Select one known text from the target author
Step 3: Concatenate the rest of the known texts from the target author
Step 4: Compute the compression distance from the selected text to the concatenated text
Step 5: Compute the compression distance from the texts of all the other authors to the concatenated text
Step 6: Iterate Step 2 to Step 5 till all the texts from the selected author are selected once
Step 7: Iterate Step 1 to Step 6 till all the authors are selected once
Step 8: Compare the distribution of the compression distances from a selected known text to the concatenated text with the distribution of the compression distances from the unknown text to the concatenated text

FIGURE 31: DESCRIPTIVE ANALYSIS PROCEDURE OF THE DESIGNED PARAMETER-BASED MODEL

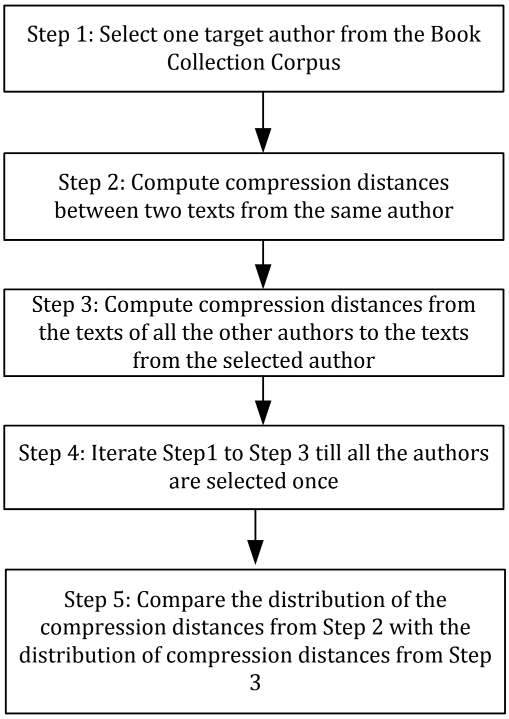

Step 1: Select one target author from the Book Collection Corpus
Step 2: Compute compression distances between two texts from the same author
Step 3: Compute compression distances from the texts of all the other authors to the texts from the selected author
Step 4: Iterate Step1 to Step 3 till all the authors are selected once
Step 5: Compare the distribution of the compression distances from Step 2 with the distribution of compression distances from Step 3

FIGURE 32: DESCRIPTIVE ANALYSIS PROCEDURE OF THE DESIGNED DISTRIBUTION-BASED MODEL

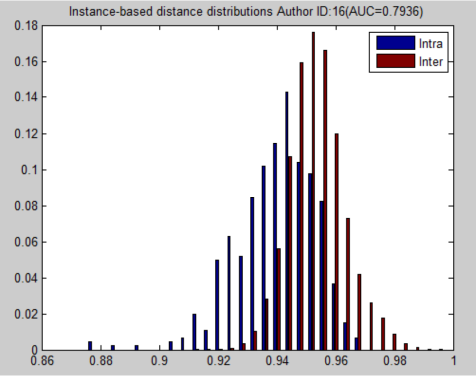
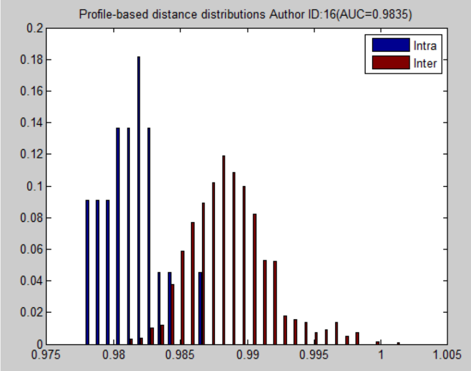
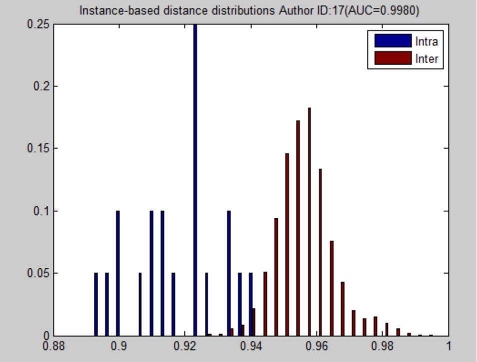
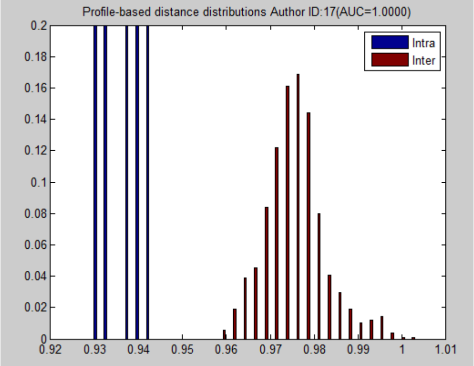
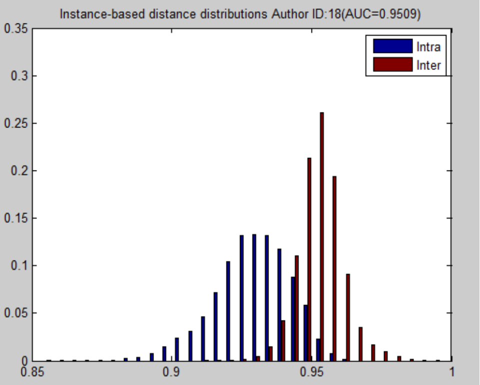
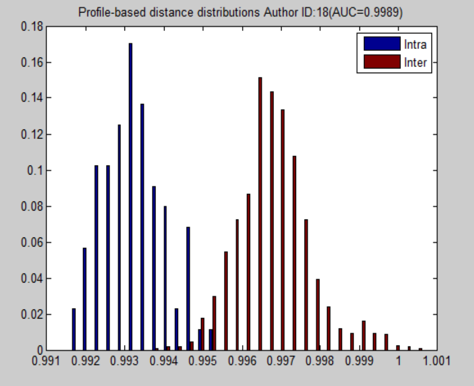
Author ID	Instance Based Distance	Profile Based Distance
16		
17		
18		

FIGURE 33: COMPARISON OF TWO DIFFERENT DATA SAMPLING APPROACHES

4.3 PARAMETER-BASED MODEL

The name of this model is derived from the decision threshold. The decision of the label 'YES' is mainly based on two statistic parameters: mean and standard deviation. The decision rule is as follows:

$$\mu'<=\mu +a\sigma$$

where μ' represents the mean of the compression distance vector of the unknown text, μ is the mean of the compression distance vector of the target class, σ is the standard deviation of the compression distance of the target class, with a factor a.

4.3.1 PARAMETER SETTINGS AND TUNING

The parameter *a* is the factor of *σ* which is a real number that is no smaller than 1 (*a* ≥1, a∈ R). In this experiment, *a* is experimented with value 1 and 2.The parameter *q* denotes the number of small texts that each given text generates. The value of *q* is tuned by several trials (*q*=2, 3, 4, 5). According to the comparison of the performances, the best result was gained when *q*=2.

4.3.2 RESULTS

Applying the Parameter-based Model to the Dataset R ten times with *a*=1 and *q*=2, the best F1 is from *CosS* compression measure. Compared with *NCD*, *CLM* and *CDM*, *CosS* has a low false positive rate. Applying the model with *a*=1 and *q*=2 to the Dataset R ten times, the best performance is again from the *CosS* measure. With regard to the other compression measures, the common problem is the relatively high false positive rate. The corresponding performance can be found in the Table 15 and Table 16. Another interesting finding is that the results of *CLM* and *CDM* are very similar.

TABLE 15: THE RESULT OF APPLYING THE MODEL TO THE DATASET R (Q=2)

Compression measure	False Positives	False Negatives	True Positives	True Negatives
***NCD* (a=2)**	17.6	5.9	44.1	32.4
***CLM* (a=2)**	11.3	7.8	42.2	38.7
***CosS* (a=2)**	2.8	8.5	41.5	47.2
***CDM* (a=2)**	10.3	7.7	42.3	39.7
***NCD* (a=1)**	9.1	7.1	42.9	40.9
***CLM* (a=1)**	4.8	9.2	40.8	45.2
***CosS* (a=1)**	0.4	12.9	37.1	49.6
***CDM* (a=1)**	4.8	8.4	41.6	45.2

TABLE 16: THE PERFORMANCE OF APPLYING THE MODEL TO THE DATASET R (Q=2)

Compression measure	Accuracy	Precision	Recall	F1
***NCD* (a=2)**	0.7650	0.7157	**0.8820**	0.7894
***CLM* (a=2)**	0.8090	0.7890	0.8440	0.8413
***CosS* (a=2)**	**0.8870**	0.9369	0.8300	**0.8793**
***CDM* (a=2)**	0.8200	0.8052	0.8460	0.8241
***NCD* (a=1)**	0.8380	0.8254	0.8580	0.8406
***CLM* (a=1)**	0.8600	0.8954	0.8160	0.8530
***CosS* (a=1)**	0.8670	**0.9899**	0.7420	0.8475
***CDM* (a=1)**	0.8680	0.8975	0.8320	0.8625

Comparing the results obtained from experiments with *a*=2 and *a*=1, *CLM* and *CosS* have improved performance when *a*=2. The other compression measures have shown slightly better performance when *a*=1. As has explained before, the recall can be traded off with precision. High false positives can contribute to high recall (more real positives captured), which would result in low precision (more wrong judgments).

4.4 DISTRIBUTION-BASED MODEL

Distribution-based Model compares two data samples, and the decision of 'YES' is made when the two samples are from the same continuous distribution. When the hypothesis of the same continuous distribution is accepted, the unknown text will be labeled with 'YES'.

4.4.1 Parameter settings and tuning

Same as the Parameter-based Model, the value of *q* is tuned by several trials (*q*=2, 3, 4, 5). Comparing the performances, the best result was gained when *q*=2.

4.4.2 Results

The result of applying the Distribution-based Model to the Dataset R is shown in the Table 17 and Table 18. Different from the Parameter-based Model, the Distribution-based Model has many more false negatives and consequently the performance is much worse.

Table 17: The Result of Applying the Model to the Dataset R (q=2)

Compression measure	False Positives	False Negatives	True Positives	True Negatives
NCD	0	46.4	0.6	50
CLM	0.5	26.9	23.1	49.5
CosS	0.5	19.5	30.5	49.5
CDM	0.3	25.1	24.9	49.7

Table 18: The Performance of Applying the Model to the Dataset R (q=2)

Compression measure	Accuracy	Precision	Recall	F1
NCD	0.5360	1	0.072	0.1333
CLM	0.7260	0.9793	0.4620	0.6245
CosS	0.8000	0.9850	0.6100	**0.7516**
CDM	0.7460	0.9883	0.4980	0.6601

4.5 Naïve Approach

Naïve approach is a two-class classification model with the basic idea of one-nearest neighbor. Different from the one-class classification models, the Book Collection Corpus has two functions in the two-class classification models. First, the same as the function in the one-class classification, the Book Collection Corpus is used to train the model. Second, as the representation of outlier class is essential, the Book Collection Corpus acts as the outlier class as well.

Each time Dataset R is created, it is applied to the Naive Approach, and its corresponding performance measures are recorded. This process is iterated ten times (it equals to run 1000 problems on this model). The average of each performance measure is summarized in the Table 19. The best F1 performance was from *CDM*. Overall, the performances of all the four compression measures are outstanding.

Table 19: The Result of Applying Dataset *R* to the Naive Approach

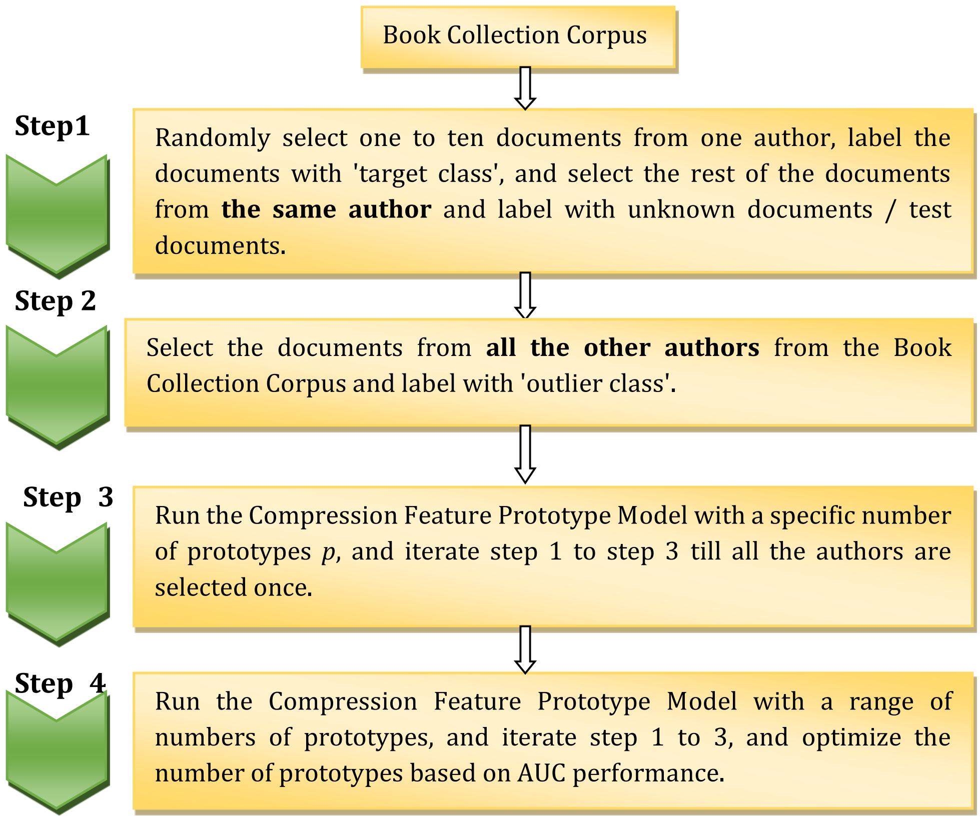
Compression measure	Accuracy	Precision	Recall	F1
NCD	0.9360	0.9867	0.8840	0.9320
CLM	0.9400	0.9851	0.8940	0.9369
CosS	0.9430	**0.9936**	0.8920	0.9395
CDM	**0.9440**	**0.9936**	**0.8940**	**0.9407**

4.6 COMPRESSION FEATURE PROTOTYPE MODEL

Compression Feature Prototype Model selects prototypes from the outlier class, and compares the compression distances from the target class to the prototypes with the compression distances from the outlier class to the prototypes. A classifier is trained to separate the target class from the outlier class.

4.6.1 PARAMETER TUNING

The number of prototypes needs tuning. It is tuned based on the following algorithm, which is described in the Figure 34. The optimized number of prototypes is 200, and C is 10,000.



Randomly select one to ten documents from one author, label the documents with 'target class', and select the rest of the documents from **the same author** and label with unknown documents / test documents.

Select the documents from **all the other authors** from the Book Collection Corpus and label with 'outlier class'.

Run the Compression Feature Prototype Model with a specific number of prototypes *p*, and iterate step 1 to step 3 till all the authors are selected once.

Run the Compression Feature Prototype Model with a range of numbers of prototypes, and iterate step 1 to 3, and optimize the number of prototypes based on AUC performance.

FIGURE 34: TUNING THE NUMBER OF PROTOTYPES

The Dataset R was generated ten times and applied to the Compression Feature Prototype Model. The result is illustrated in Table 20. The high recall and low precision indicate that both compression metrics tend to label 'Positive' in most problems.

TABLE 20: RESULT OF APPLYING DATASET R TO THE COMPRESSION FEATURE PROTOTYPE

Compression measure	Accuracy	Precision	Recall	F1
CosS	0.5420	0.5221	**0.9940**	0.6846
CDM	**0.6630**	**0.6061**	0.9320	**0.7937**

4.7 CHARACTER N-GRAM MODEL

Table 21 describes the properties of character *n*-gram features. For the character bigrams, all the features are selected, while for the character trigrams 3500 features are selected based on their relative frequencies. The sum of the relative frequencies of the selected 3500 features is 0.9515. The results of the application of the Character N-Gram Model to the Dataset V are shown in Table 22 and Table 23. LIBSVM is used to classify the two classes. LIBSVM is a library for Support Vector Machine which is developed to facilitate the SVM applications (Chang & Lin, 2011). Parameter *C* (regularization parameter) is tuned by optimizing the F1 measure. As a consequence, *C*=12743 has the best F measure for the trigrams and *C*=1 for the bigrams.

TABLE 21: CHARACTER N-GRAM FEATURE PROPERTIES

Character n-grams	N=2(Bigrams)	N=3 (Trigrams)
Number of Features	4339	36261
Number of selected features	4339	3500
Cumulative frequency percentage of the selected features to the complete features	100%	95.15%

In order to validate the model, a separate Dataset V is prepared in a similar way as the Dataset R. The only difference is that the five PAN authors were excluded. As a consequence, for each Dataset V there are 46 authors with 92 problems (46 YES and 46 NO) to solve.

TABLE 22: RESULTS OF APPLYING THE CHARACTER N-GRAM MODEL TO THE DATASET V (CLASSIFIER SVM)

N-GRAM	False Positives	False Negatives	True Positives	True Negatives
N=3	0	16	30	46
N=2	0	22	24	46

TABLE 23: PERFORMANCE OF APPLYING THE CHARACTER N-GRAM MODEL TO THE DATASET V (CLASSIFIER SVM)

N-GRAM	Accuracy	Precision	Recall	F1
N=3	0.8261	**1**	0.6522	0.7888
N=2	0.7609	**1**	0.5217	0.6847

4.8 COMPARISON

TABLE 24: THE COMPARISON OF THE PERFORMANCE OF THE DESIGNED MODELS

	Name	Feature	Classifier	F1-measure (Compression measure)	Test dataset
One-Class	Parameter-based Model	Compression features	None	0.8793 (CosS)	R
	Distribution-based Model	Compression features	None	0.7516(CosS)	R
Two-Class	Naive Approach	Compression features	None	0.9407 (CosS)	R
	Compression Feature Prototype Model	Compression features	LESS Classifier	0.7937 (CDM)	R
	Character N-Gram Model	Character trigrams	LIBSVM	0.7888	V

The comparison of F1 performances between the designed models can be found in the Table 24. The best performance is from the Naive Approach, which is designed as a two-class classification model. A major criticism of the Naïve Approach is that it is not robust. Among the two-class classification models, models using compression features are at least as good as the Character N-Gram Model using character trigrams or character bigrams. Among the one-class classification models, the Parameter-based Model has better performance as expected.

CHAPTER 5 MODEL EVALUATION

5.1 EVALUATION WITH ENRON EMAIL CORPUS

The given training dataset of PAN Contest 2011 in the track of authorship identification is used to evaluate the designed one-class classification approaches. This dataset was derived from the corporate emails of Enron Corporation. In total, 36 authors are included in the email dataset. The average length of each email is 331 characters, in the range from 18 to 212478 characters. Some comparisons of the intra-compression distances and inter-compression distances of the email corpus are described in Figure 35. As is shown in the Figure 35, neither the profile-based distances nor the instance-based distances can separate the emails written by the same author from the emails written by the other authors.

In order to know if the short length of an email is the critical factor that makes the intra-compression distances inseparable from the inter-compression distances, the emails from the same author were first merged into one big text, and then the text was split into pieces of small texts with a text length of 500 characters each. Three authors, each of whom has only one short email were removed from the corpus. The average of the AUC of the instance-based distances is 0.5477, a number which is equal to random guessing; the average of the AUC of the profile-based distances is 0.9335. The full list of the AUC performance can be found in the appendix D, and a sample of the comparison of the intra-compression distances and inter-compression distances can be found in the Figure 36. The profile-based distances showed a substantial improvement compared to the instance-based distances. This supports the finding that the profile-based distances can better separate the texts from the same from the texts written by other authors, when the amount of each text is sufficient. This implies that the designed Parameter-based Model may work well for the email corpus, while the Distribution-based Model will probably not work.

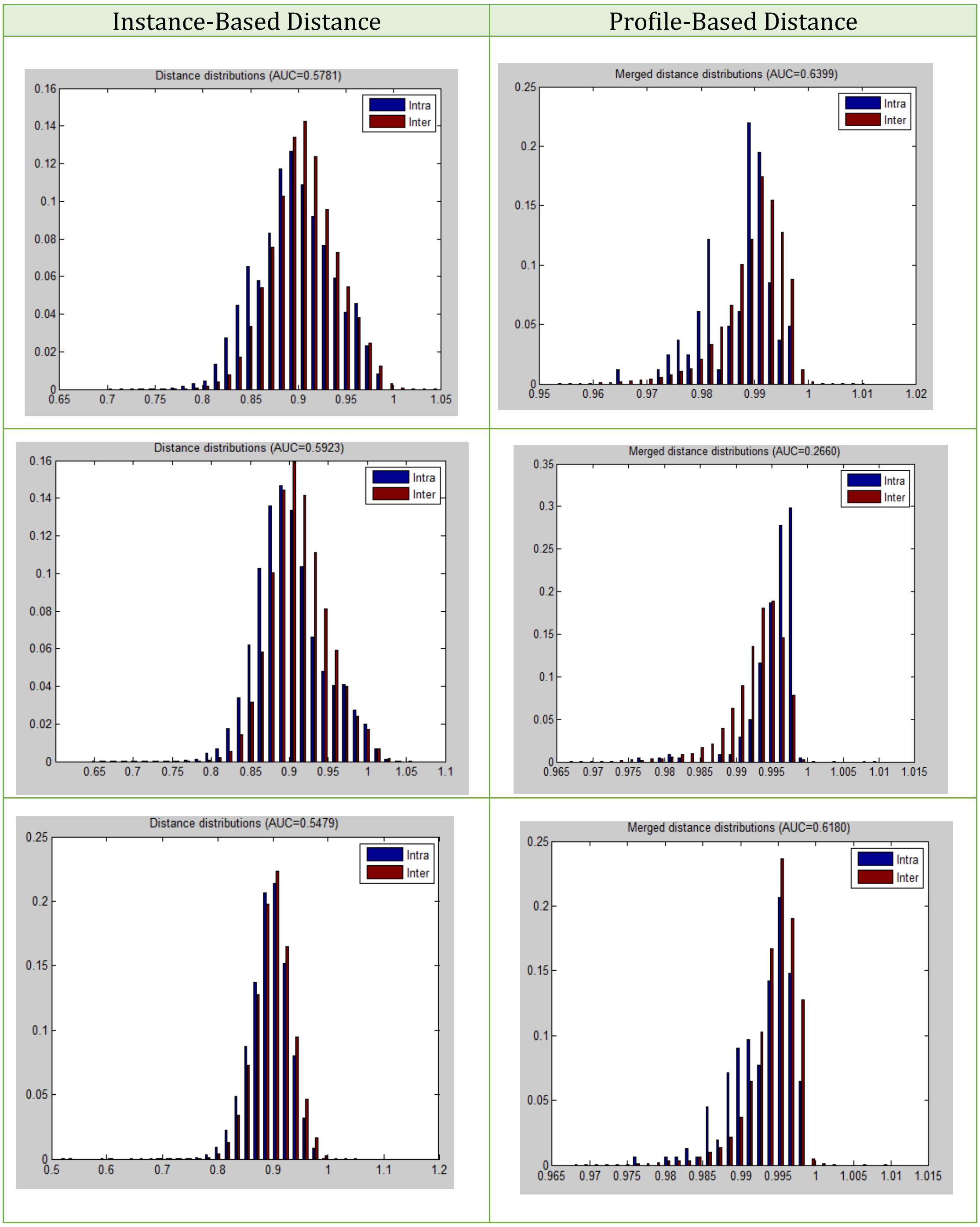


FIGURE 35: COMPARISON OF THE INSTANCE-BASED DISTANCE WITH THE PROFILE-BASED DISTANCE OF THE ENRON EMAIL CORPUS

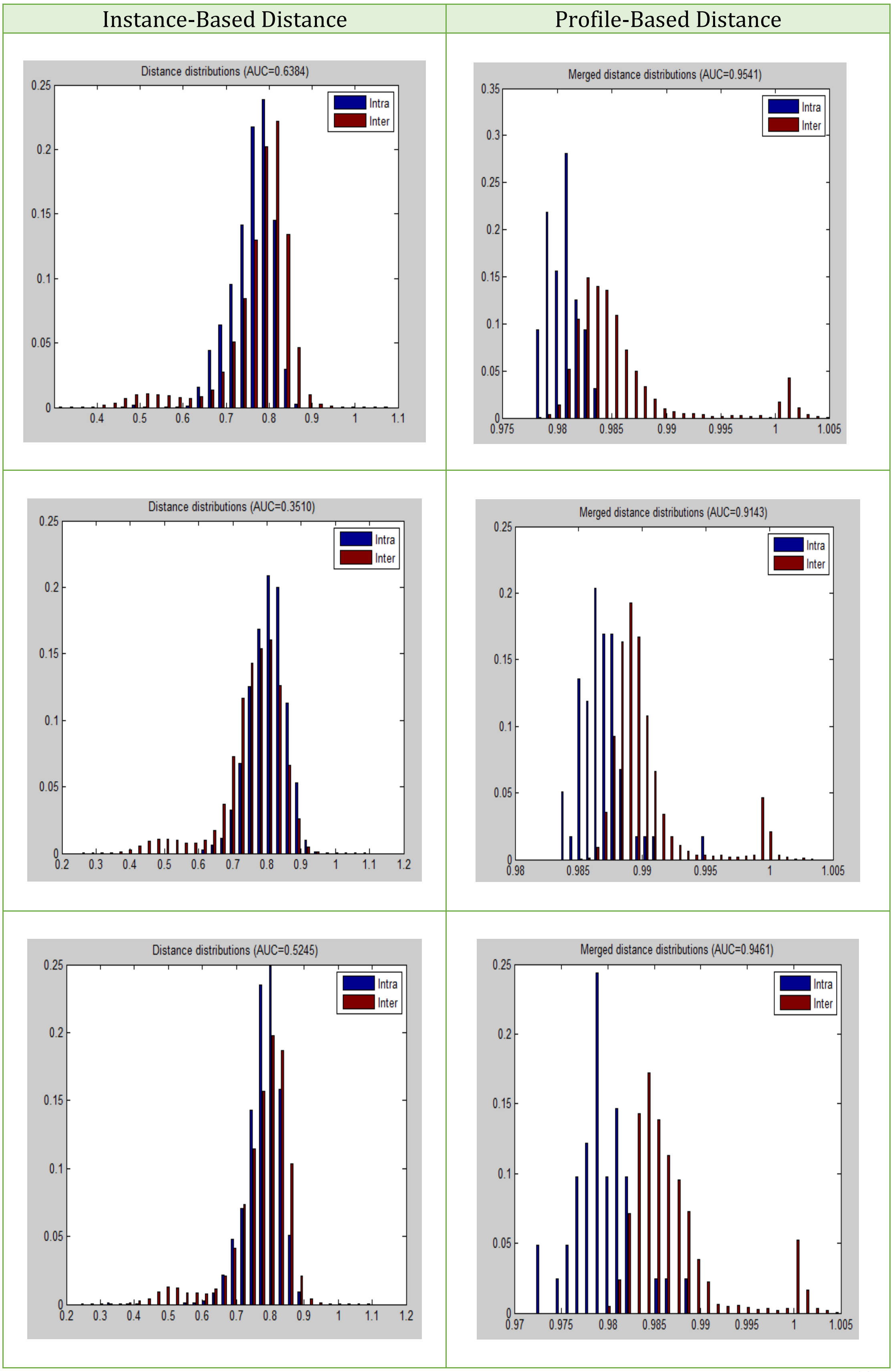


FIGURE 36: COMPARISON OF THE INSTANCE-BASED DISTANCE WITH THE PROFILE-BASED DISTANCE OF THE ENRON EMAIL CORPUS (500 CHARACTERS EACH TEXT)

A small test dataset similar to Dataset R and V was created to evaluate the performance of the Parameter-based Model. In order to create the small separate test dataset, the emails from the same author were merged into one text, and then the text was split into small texts around 1000 characters for each small text. Only 64 problems were in the dataset. The results are shown in Table 25. The decision rule is as follows:

$$\mu'<=\mu+a\sigma$$

where μ' is the mean of the compression distances from the single unknown text to the concatenated texts, μ is the mean of the compression distances from the known texts to the concatenated texts, and a is a factor of σ. The results have shown a better performance gained from the compression measure *CosS*. Additionally, the results indicate that the selection of a has a strong impact on the prediction. Different from the Book Collection Corpus, the Enron Email Corpus has the worst performance when a is equal to 2.

Table 25: Performance of the parameter-based Model on the email corpus

Compression measure	Accuracy	Precision	Recall	F-Measure
***CosS* (*a*=0.5)**	**0.73**	**0.70**	**0.84**	**0.76**
***CosS* (*a*=1)**	0.70	0.66	0.84	0.74
***CosS* (*a*=2)**	0.58	0.55	0.88	0.67
***CDM* (*a*=0.5)**	0.61	0.57	0.88	0.69
***CDM* (*a*=1)**	0.60	0.56	0.88	0.68
***CDM* (*a*=2)**	0.56	0.54	0.88	0.67

5.2 Evaluation with PAN's Dataset

The second goal of this research is to participate in the PAN Contest 2013 under the track of authorship verification. Hence, the validity of the designed model can be evaluated with the datasets provided by the PAN Organization. PAN Organization has provided a reference dataset, which consists of 10 English problems, 20 Greek problems, and 5 Spanish problems. The objective of the reference dataset provided by PAN is to evaluate the performance of the models developed by participants by themselves. Another much larger dataset which is not released to the public is used to test the submitted models. Only the Instance-based Compression Model which was developed first has been submitted to the PAN Contest 2013 due to the time limitation.

5.1.1 One-class Classification Model

The results of the *Parameter-based Model* for each language are summarized in Table 26. The table shows that this model can achieve relatively good results in English and in Spanish, whereas the result of the Greek problems is not desirable. When applying the model to solve the 10 English problems, *NCD* has three false positives, *CLM* and *CDM* has one false positive, while *CosS* has one false negative label. With regard to 5 Spanish problems, the Parameter-based Model correctly predicts all the five labels, regardless of the selection of the compression measures.

TABLE 26: REPRESENTATION OF THE RESULT WITH THE PERFORMANCE MEASURES (A=2)

Compression measure	Accuracy	Precision	Recall	F-Measure
NCD (English)	0.70	0.63	1.00	0.77
CLM (English)	0.90	0.83	1.00	0.91
CosS (English)	0.90	1.00	0.80	0.89
CDM (English)	0.90	0.83	1.00	0.91
NCD (Greek)	0.50	0.50	0.90	0.64
CLM (Greek)	0.55	0.53	0.80	0.64
CosS (Greek)	0.65	0.71	0.50	0.59
CDM (Greek)	0.55	0.53	0.80	0.64
NCD (Spanish)	1	1	1	1
CLM (Spanish)	1	1	1	1
CosS (Spanish)	1	1	1	1
CDM (Spanish)	1	1	1	1

According to the results obtained from Model Design Chapter, only *CosS* and *CDM* are selected to evaluate the model. When applying the model to solve the ten English problems, both of them have two false negatives. With regard to five Spanish problems, the *CosS* has one false negative label, while *CDM* has two false negative labels. The same as the Parameter-based Model, Distribution-based Model fails to solve the Greek problems.

TABLE 27: RESULTS OF THE DISTRIBUTION-BASED MODEL

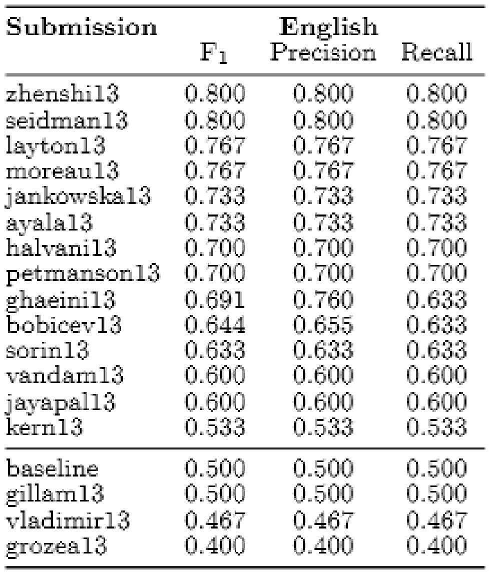
Compression measure	False Positives	False Negatives	True Positives	True Negatives
CosS (10 English)	0	2	3	5
CDM (10 English)	0	2	3	5
CosS (5 Spanish)	0	1	2	2
CDM (5 Spanish)	0	2	1	2
CosS (30 Greek)	0	9	1	10
CDM (30 Greek)	0	9	1	10

TABLE 28: PERFORMANCE OF THE DISTRIBUTION-BASED MODEL

Compression measure	Accuracy	Precision	Recall	F-Measure
CosS (10 English)	0.80	1.00	0.60	0.75
CDM(10 English)	0.80	1.00	0.60	0.75
CosS (5 Spanish)	0.80	1.00	0.67	0.80
CDM (5 Spanish)	0.60	1.00	0.33	0.50
CosS (30 Greek)	0.55	1.00	0.10	0.18
CDM (30 Greek)	0.55	1.00	0.10	0.18

5.1.2 TWO-CLASS CLASSIFICATION MODEL

For all the two-class classification models, the outlier class representation is of great importance. Thus, it only applies to the English problems in this research due to the data collection procedure. When applying the Naive Approach to the PAN reference dataset, *CDM* has only one false positive. When applying the Two-Class Compression Prototype Model, the average error rate is 26%, and the Bootstrapping Approach has an average error rate 16%. These three models were submitted to the PAN Contest, and the evaluation result is shown in the Figure 37.



Submission	English		
	F_1	Precision	Recall
zhenshi13	0.800	0.800	0.800
seidman13	0.800	0.800	0.800
layton13	0.767	0.767	0.767
moreau13	0.767	0.767	0.767
jankowska13	0.733	0.733	0.733
ayala13	0.733	0.733	0.733
halvani13	0.700	0.700	0.700
petmanson13	0.700	0.700	0.700
ghaeini13	0.691	0.760	0.633
bobicev13	0.644	0.655	0.633
sorin13	0.633	0.633	0.633
vandam13	0.600	0.600	0.600
jayapal13	0.600	0.600	0.600
kern13	0.533	0.533	0.533
baseline	0.500	0.500	0.500
gillam13	0.500	0.500	0.500
vladimir13	0.467	0.467	0.467
grozea13	0.400	0.400	0.400

FIGURE 37: THE RESULT OF THE PAN CONTEST (ENGLISH)

The performance of the Character N-Gram Model is only evaluated with the PAN's reference data (ten English problems). The LIBSVM has no false positive prediction but two false negatives (both character trigram and character bigram) in the ten English problems.

TABLE 29: THE RESULT OF CHARACTER N-GRAM MODEL (10 ENGLISH PROBLEMS)

Classifier	Character 2-Gram Model		Character 3-Gram Model	
	False Positive	False Negative	False Positive	False Negative
LIBSVM	0	2	0	2

CHAPTER 6 CONCLUSION AND RECOMMENDATION

6.1 OVERVIEW OF THE RESEARCH

The entire research adopts a supervised learning approach. In total, five different models have been designed and evaluated in this research: two *one-class classification* models and three *two-class classification* models. Comparing two *one-class classification* models, the Parameter-based Model has a better performance than the Distribution-based Model. In terms of the *two-class classification* models, the Naive Approach outperforms the rest models. Moreover, as a relatively new compression measure, *CosS* generally has a better performance than other compression measures in this research. The first objective of this research is to design several authorship verification models with a good performance. This objective has been well achieved. Additionally, since the timeline of the PAN Contest is in alignment with this research, the designed *two-class classification* models were submitted to participate in the contest. The results of the designed models are desirable compared to some similar research (Lambers & Veenman, 2009; Grieve, 2007; Luyckx & Daelemans, 2008). The models submitted to the PAN Contest received the best evaluation result among the 18 teams in the English task.

The answers to the research questions are summarized as follows.

Q1: What are the existing methods that have been used to solve the similar problem?

As has been said in the first chapter, the root of the authorship analysis is stylometry study. Hence, the stylometric features are the dominant features when conducting an authorship analysis. On the contrary, compression features are relatively new and are not widely adopted. However, compression features have been shown to be promising. Both one-class classification approach and two-class classification approach have been used to solve the authorship attribution problems. Likewise, both profile-based approach as well as instance-based approach has been adopted to solve the problem, while the instance-based approach is more popular. In terms of the computation techniques, different kinds of machine learning techniques have been implemented by researchers. The Support Vector Machine is one of the most popular classification algorithms in the authorship attribution study.

Q2: How the models should be designed?

This research is designed to be an exploratory research. The overview of the designed model can be found in Table 30. In terms of robustness, only Compression Feature Prototype Model and Character N-Gram Model are supposed to be robust, whereas the other three models may not be very robust. However, the two-class classification approach requires a representation of the outlier class, which necessitates outlier data collection. Therefore, one-class classification has relatively less cost.

Table 30: Overview of the models

Model	Profile-based approach / Instance-based approach	One-class classification/two-class classification	Classifier/ Decision rule
Parameter-based Model	Profile-based Approach	One-class classification	$\mu'<=\mu +a\sigma$
Distribution-based Model	Instance-based Approach	One-class classification	Two-sample Kolmogorov-Smirnov Test
Naïve Approach	Instance-based Approach	Two-class classification	K-nearest neighbor
Compression Feature Prototype Model	Instance-based Approach	Two-class classification	*LESS* classifier
Character N-Gram Model	Instance-based Approach	Two-class classification	SVM classifier

Q3: How is the validity of the model designed?

All the five predictive models designed are valid and effective when they are applied to the Book Collection Corpus. The Parameter-based Model is free of the outlier data collection, and hence it can be used to solve problems from other languages. Evaluating the Parameter-based Model with the Enron Email Corpus has shown a potential of this model to solve authorship verification problems of emails.

6.2 Findings

As is shown in Table 24, the models utilizing compression features are at least as good as the Character N-Gram Model when applied to the Book Collection Corpus. This indicates that compression features can also solve the authorship attribution effectively. Moreover, this research has found that the profile-based approach can better separate the intra-compression distances from the inter-compression distances, i.e. compression distances of the same author from the compression distances between different authors. This requires a sufficient amount of characters for each text. In this research the minimum length of each text experimented is 100 characters (around 180 words). With regard to the Parameter-based Model, it is found that the selection of parameter *a* is of great importance. When *a* is 2, the model works very well to solve the problems derived from the Book Collection Corpus. However, when *a* is 2, the model has the lowest performance to solve the problems generated from the Enron Email Corpus.

6.3 LIMITATIONS

In the Book Collection Corpus, normally one author only has one book. Thus, the variety of the contents of different books may contribute to differentiating different authors instead of authors' writing styles. Additionally, inaccessibility of the PAN's evaluation dataset makes it impossible to evaluate the one-class classification models and the Character N-Gram Model with their evaluation dataset. Moreover, only books were collected, and therefore the performance of the designed models might vary significantly when solving authorship verification problems of computer-mediated messages (e.g. blogs). Though a short evaluation has been done on an email corpus, it is necessary to conduct another research to evaluate the models further with other email corpora.

These models are not very likely to implement to solve the real-life forensic authorship verification problems. First of all, 1000 words for each text are still too long. Some of the authorship verification problems are just a few sentences (e.g. a threat letter from a criminal). In terms of usability, one-class classification models are more likely to be implemented to solve the real problems. The necessity of collecting outlier data makes the two-class classification models language-dependent. In addition, the outlier data should be as close to the target class as possible, which is another constraint. However, the one-class classification models have no requirements for an outlier class representation.

6.4 RECOMMENDATION AND FUTURE RESEARCH

Some other types of documents (e.g. XML) can be used as the data source to test the performance of the designed models. For the sensitive models that are not robust, some adjustments can be done to improve their robustness. For instance, the Naïve Approach adopts the k-nearest neighbor algorithm, and the k is selected as 1 in this research. For the improvement, k can be designed as a variable instead of a constant number. One critical issue of the research problem is that there are only one to ten texts of each author. If k is 5, then when there are two known texts, the unknown text will be always classified as the outlier class. However, if the k is a variable which is equal to the number of the available known texts, this problem can be solved and the robustness is supposed to be improved to some extent. Concerning the Parameter-based Model, the length of each text and the choice of the factor a are two crucial parameters to make the model work properly. According, what is the minimum length of a text that is required to have the model work effectively and how to set the factor a are two interesting points to be studied further.

REFERENCES

Alamdari, A. R., & Guyon, I. (2006). *Quick Start Guide for CLOP.* Retrieved from CLOP: http://clopinet.com/CLOP/QuickStart.pdf

Andress, J., & Winterfeld, S. (2011). *CYBER WARFARE: Techniques, Tactics and Tools for Security Practitioners.* SYNGRESS.

Baayen, H., van Halteren, H., & Tweedie, F. (1996). Outside the cave of shadows: using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing, 11*(3), 121-132.

Bozkurt, İ. N., Bağlıoğlu, Ö., & Uyar, E. (2007). *Authorship Attribution .* Retrieved 7 14, 2013, from The Department of Computer Science at Duke University: https://www.cs.duke.edu/~ilker/papers/conference/iscs07.pdf

Chaitin, G. (1969). On the length of programs for computing finite binary sequences: statistical considerations. *Journal of the ACM, 16*(1), 145-159.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 1:27.

Chen, X., Francia, B., Li, M., McKinnon, B., & Seker, A. (2004). Shared Information and Program Plagiarism Detection. *IEEE TRANSACTIONS ON INFORMATION THEORY, 50*(7), 1545-1551.

Cilibrasi, R., & Vitányi, P. M. (2005). Clustering by Compression. *IEEE TRANSACTIONS ON INFORMATION THEORY, 51*(4), 1523-1545.

CLEARY, J. G., & WITTEN, I. H. (1984). Data Compression Using Adaptive Coding and Partial String Matching. *IEEE TRANSACTIONS ON COMMUNICATIONS, COM-32*, 396-402.

Daelemans, W., & van den Bosch, A. (2005). *MEMORY-BASED LANGUAGE PROCESSING.* Cambridge: Cambridge University Press.

Damashek, M. (1995). Gauging Similarity with n-Grams: Language-Independent Categorization of Text. *American Association for the Advancement of Science, 267*(5199), 843-848.

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. *First workshop on Multiple Classifier Systems. LNCS Volume 1857* (pp. 1-15). Springer.

Duin, R., Juszczak, P., Paclik, P., Pekalska, E., Ridder, D. d., Tax, D., & Verzakov, S. (2007, August). *PRTools4 A Matlab Toolbox for Pattern Recognition.* Retrieved from PRTools: http://prtools.org/files/PRTools4.1.pdf

Eder, M. (2010). *DH2010: Does Size Matter? Authorship Attribution, Small Samples, Big Problem.* Retrieved 2 8, 2012, from DH2010:DIGITAL HUMANITIES:

http://dh2010.cch.kcl.ac.uk/academic-programme/abstracts/papers/html/ab-744.html

Escalante, h. J. (2011). EPSMS and the Document Occurrence Representation for Authorship Identification. *IN Notebook for PAN at CLEFF 2011.* Amsterdam.

Escalante, H. J., Montes, M., & Sucar, E. (2010). Ensemble Particle Swarm Model Selection. *World Congress on Computational Intelligence* (pp. 1814-1821). Barcelona: IEEE.

Escalante, H. J., Montes, M., & Sucar, L. E. (2009). Particle Swarm Model Selection. *Journal of Machine Learning Research*, 405-440.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., & Wang, X.-R. (2008, 4 24). LILIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research 9*, 1871-1874. Retrieved from LIBLINEAR -- A Library for Large Linear Classification.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* Cambridge: MIT Press.

Grieve, J. (2007). Quantitative Authorship Attribution: An Evaluation of the Techniques. *Literary and Linguistic Computing, 22*(3).

Halteren, H. v. (2004). Linguistic Profiling for Author Recognition and Verification. *ACL '04 Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics.* Stroudsburg.

HOOVER, D. L. (2003). Another Perspective on Vocabulary Richness. *Computers and the Humanities , 37*, 151–178.

Iqbal, F., Fung, B. C., Khan, L. A., & Debbabi, M. (2010). *E-mail Authorship Verification for Forensic Investigation.* Retrieved 12 10, 2012, from National Cyber-Forensics and Training Alliance CANADA: http://www.ncfta.ca/papers/IKFD10sac.pdf

Juola, P. (2006). Authorship Attribution. *Information Retrieval, 1*(3), 233-334.

Kanaris, I., Kanaris, K., Houvardas, I., & Stamatatos, E. (2007). WORDS VS. CHARACTER N-GRAMS FOR ANTI-SPAM FILTERING. *International Journal on Artificial Intelligence Tools, 16*, 1047-1067.

Kati. (2012, May 22). *Bookboon.com: 600+ Open access textbooks*. Retrieved from bookboon.com: http://bookboon.com/blog/2012/05/bookboon-com-800-open-access-textbooks/

Keogh, E., Lonardi, S., & Ratanamahatana, C. A. (2004). Towards Parameter-Free Data Mining. *In: Proceedings of the 10th ACM SIGKDD International conference on Knowledge Discovery and Data Mining*, (pp. 206-215).

Kolmogorov, A. (1965). Three approaches to the quantitative definition of information. *Problemy Peredachi Informatsii, 1*(1), 1-7.

Koppel, M., & Schler, J. (2004). Authorship Verification as a One-Class Classification Problem . *The 21st International Conference on Machine Learning.* Banff.

Koppel, M., Schler, J., & Argamon, S. (2009). Computational Methods in Authorship Attribution . *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY*, (pp. 60(1):9–26).

Kourtis, I., & Stamatatos, E. (2011). Authorship Identification Using Semi-supervised Learning. *In: Notebook for PAN at CLEFF 2011.* Amsterdam.

Lambers, M., & Veenman, C. J. (2009). Forensic Authorship Attribution Using Compression Distances to Prototypes. *In Computational Forensics, Lecture Notes in Computer Science, Volume 5718*, (pp. 13-14).

Lee, S.-I., Lee, H., Abbeel, P., & Ng, A. Y. (2006). Efficient L1 Regularized Logistic Regression. *American Association for Artificial Intelligence*, 401-408.

Li, M., Badger, J. H., Xin Chen, S. K., & Zhang, H. (2001). An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *BIOINFORMATICS, 17*(2), 149-154.

Li, M., Chen, X., Li, X., Ma, B., & Vitányi, P. M. (2004). The Similarity Metric. *IEEE TRANSACTIONS ON INFORMATION THEORY, 50*(12), 3250-3264.

Luyckx, K., & Daelemans, W. (2008). Authorship Attribution and Verification with Many Authors and Limited Data. *The 22nd International Conference on Computational Linguistics*, (pp. 513-520). 513-520.

Massey, F. J. (1951). The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association, 46*(253), 68-78.

Mikros, G. K., & Perifanos, K. (2011). Authorship identification in large email collections:Experiments using features that belong to different linguistic levels. *IN: Notebook for PAN at CLEFF.* Amsterdam.

Miller, L. H. (1956). Table of Percentage Points of Kolmogorov Statistics. *Journal of the American Statistical Association, 51*(273), 111-121.

MOFFAT, A. (1990). Implementing the PPM Data Compression Scheme. *IEEE TRANSACTIONS ON COMMUNICATIONS, 38*(11), 1917-1921.

Mosteller, F., & Wallace, D. L. (1963). Inference in an Authorship Problem. *Journal of the American Statistical Association, 58*(302).

Olson, D. L., & Delen, D. (2008). *Advanced Data Mining Techniques.* Berlin, Heidelberg: Springer.

Orsyth, R., & Holmes, D. (1996). Feature-finding for text classification. *Literary and Linguistic Computing, 11*(4), 163-174.

PAN. (2013). *PAN 2013*. Retrieved 1 28, 2012, from PAN 2013: http://pan.webis.de/

Potthast, M. (2012, 2 11). *Uncovering Plagiarism, Authorship and Software Misuse: PAN 2011 Results.* Retrieved from PAN@CLEFF2011 | Lab on Uncovering Authorship, Plagiarism and Social Software Misuse: http://www.uniweimar.de/medien/webis/research/events/pan-11/pan11-talks/pan11-results.pdf

Santorini, B. (1990, June). *The Penn Treebank Project*. Retrieved 2 18, 2013, from The Penn Treebank Project: http://www.cis.upenn.edu/~treebank/

Sculley, D., & Brodley, C. E. (2006). Compression and Machine Learning: A New Perspective on Feature Space Vectors. *DCC '06 Proceedings of the Data Compression Conference*, (pp. 332-332).

Shkarin, D. (2002). PPM: one step to practicality. *In Proceedings of the DATA COMPRESSION CONFERENCE.* IEEE.

Solomonoff, R. (1964). A formal theory of inductive inference Part 1. *Information and Control, 7*(1), 1-22.

Solomonoff, R. (1964). A formal theory of inductive inference Part 2. *Information and Control, 7*(2), 224-254.

Stamatatos, E. (2009). A Survey of Modern Authorship Attribution Methods. *Journal of the American Society for Information Science and Technology, 60*(3), 238-556.

Tanguy, L., Urieli, A., Calderone, B., Hathout, N., & Sajous, F. (2011). A multitude of linguistically-rich features for authorship attribution. *IN: Notebook for PAN at CLEFF 2011.* Amsterdam.

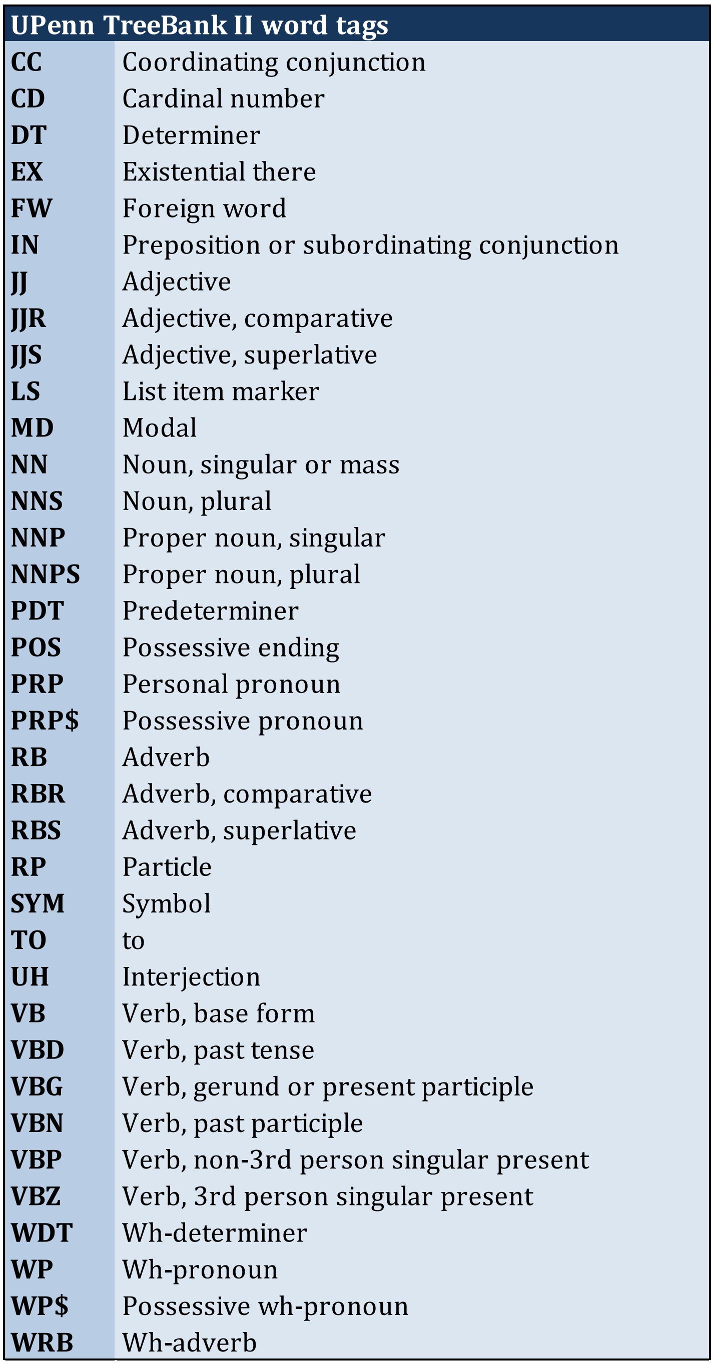
Veenman, C. J., & Tax, D. M. (2005). LESS: A Model-Based Classifier for Sparse Subspaces. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 27*(9), 1496-1500.

Veenman, C., & Li, Z. (2013). Authorship Verification with Compression Features. *IN Notebook for PAN at ClEFF 2013.* Valencia.

Yang, Y., & Perdersen, J. O. (1997). A Comparative study on feature selection in text categorization. *In International Conference on Machine Learning*, (pp. 412-420).

Zheng, R., Li, J., Chen, H., & Huang, Z. (2006). A Framework for Authorship Identification of Online Messages:Writing-Style Features and Classification Techniques. *JOURNAL OF THE AMERICAN SOCIETY INFORMATION SCIENCE TECHNOLOGY*, 378-393.

APPENDIX A PART-OF-SPEECH TAGS



UPenn TreeBank II word tags	
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possessive wh-pronoun
WRB	Wh-adverb

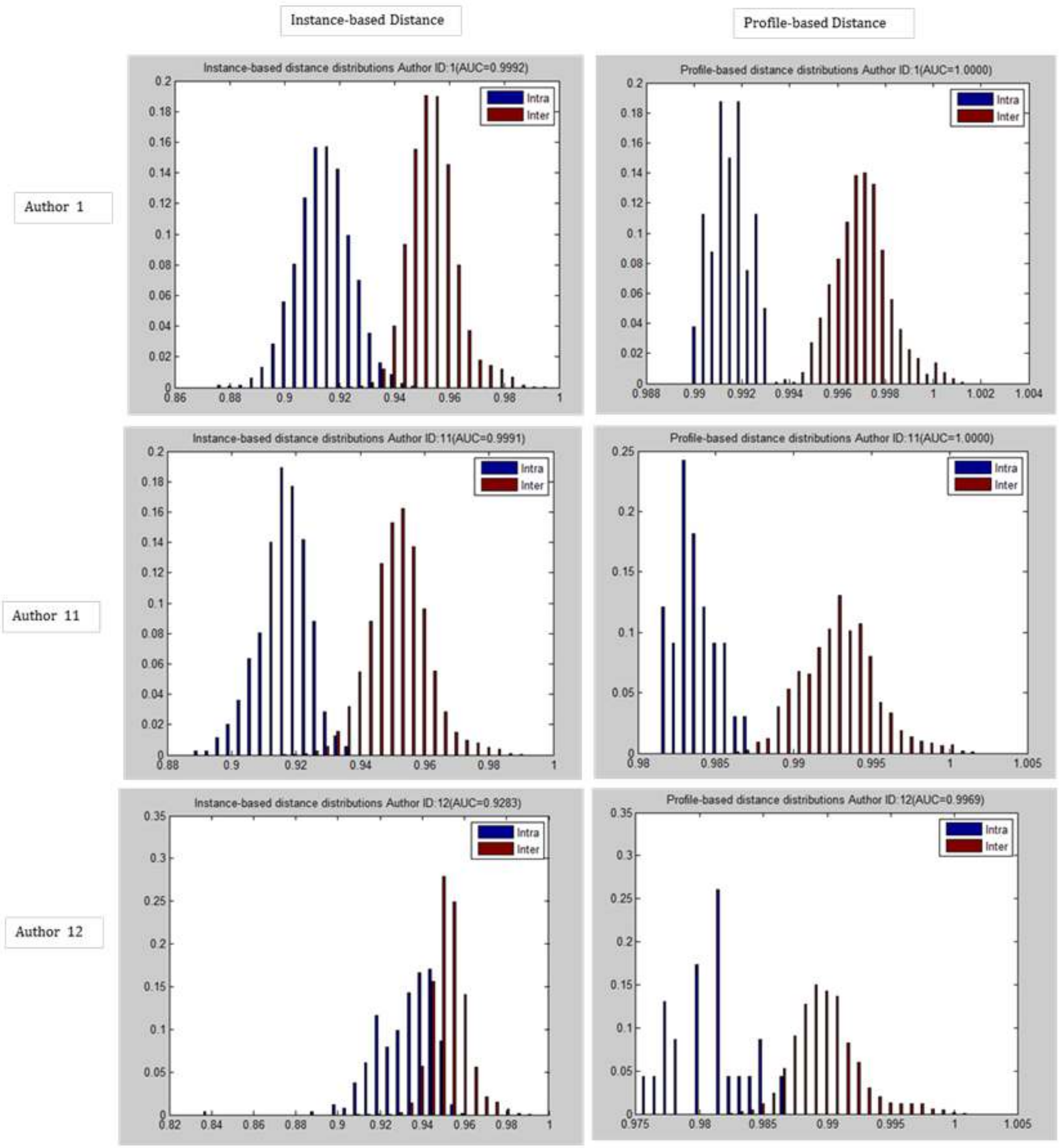
FIGURE 38 PART-OF-SPEECH TAGS (SANTORINI, 1990)

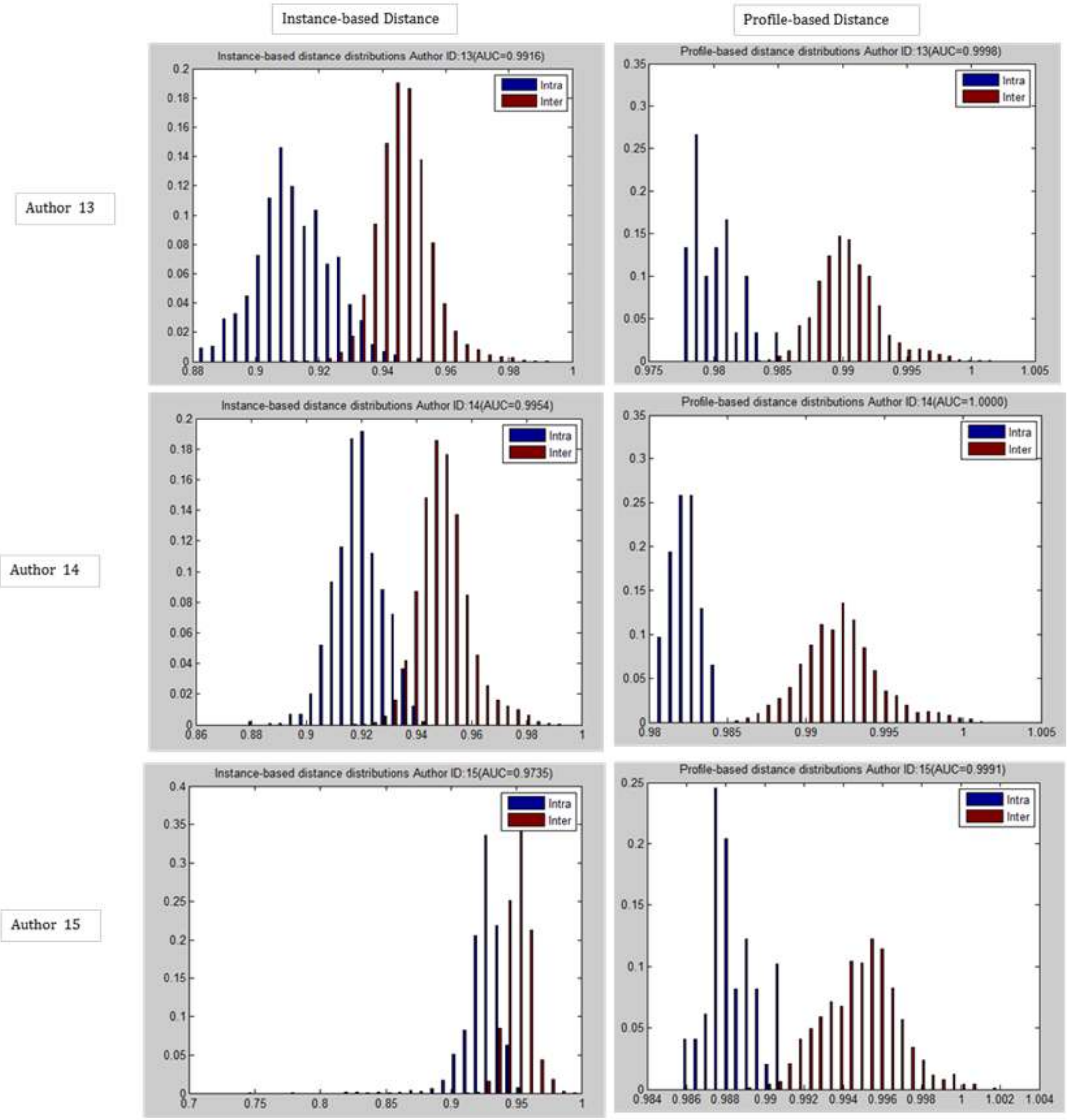
APPENDIX B AUTHOR AND BOOK LISTS OF DATA COLLECTION

Author ID	Author Name	PAN AUTHOR(Y/N)
1	C. J. Date	N
2	Christopher Fox	N
3	David Etheridge	N
4	David Haskins	N
5	Derek Atherton	N
6	Geoffrey Sampson	N
7	Hugh Darwen	N
8	Karol Kozak	Y
9	Kjell Backman	Y
10	Krister Ahlersten	N
11	Poul Klausen	N
12	Ramaswamy Palaniappan	Y
13	Richard Carter	N
14	Roger McHaney	Y
15	Simon Kendal	N
16	Tarik AI-Shemmeri	N
17	Udo Richard Franz Averweg	N
18	Valery Vodovozov	N
19	W. J. R. H. Pooler	N
20	Wasif Naeem	N
21	Weijing Wang	Y
22	William John Teahan	N
23	Peter Dybdahl Hede	N
24	Buddhi N. Hewakandamby	N
25	Amab Roy	N
26	J.C. Jones	N
27	Miltiadis A. Boboulos	N
28	Peter Moir	N
29	Graeme M. Walker	N
30	Leo Lue	N
31	Ashleigh J. Fletcher	N
32	J.E. Parker	N
33	Abdulnaser Sayma	N
34	Peter Klappa	N
35	Pal Skalle	N
36	Soren Prip Beier	N
37	Carl J. Schaschke	N
38	Romain Elsair	N
39	Vladimir Molkov	N
40	Rafael Kandiyoti	N
41	Momna Hejmadi	N
42	Graham Basten	N
43	Peter G. Nelson	N
44	Mustafa Akay	N
45	Grant Ingram	N
46	David Bakewell	N
47	J. Richard Wilson	N
48	Christopher Wood	N
49	Jeremy Ramsden	N
50	Philip Rowe	N
51	Jeremiah Rushchitsky	N
TOTAL	**51 authors**	**5 pan authors**

NO.	BOOK TITLE	AUTHOR	PAN BOOK (Y/N)
1	The transRelational Approach to DBMS Implementation	C. J. Date	N
2	Concise Notes on Data Structures and Algorithms	Christopher Fox	N
3	Java: The Fundamentals of Objects and Classes	David Etheridge	N
4	C Programming in Linux	David Haskins	N
5	Control Engineering	Derek Atherton	N
6	An introduction to Nonlinearity in Control Systems	Derek Atherton	N
7	Perl for Beginners	Geoffrey Sampson	N
8	SQL: A Comparative Survey	Hugh Darwen	N
9	An introduction to Relational Database Theory	Hugh Darwen	N
10	Large Scale Data Handling in Biology	Karol Kozak	**Y**
11	Structured Programming with C++	Kjell Backman	**Y**
12	An introduction to Matlab	Krister Ahlersten	N
13	Introduction to programming and the C# language	Poul Klausen	N
14	Biological Signal Analysis	Ramaswamy Palaniappan	**Y**
15	Digital System Design	Ramaswamy Palaniappan	**Y**
16	Electromagnetism for Electronic Engineers	Richard Carter	N
17	Understanding Computer Simulation	Roger McHaney	**Y**
18	Object Oriented Programming using C#	Simon Kendal	N
19	Object Oriented Programming using Java	Simon Kendal	N
20	Engineering Thermodynamics	Tarik AI-Shemmeri	N
21	Wind Turbines	Tarik AI-Shemmeri	N
22	Decision-making support systems	Udo Richard Franz Averweg	N
23	Electric Drive Dimensioning and Tuning	Valery Vodovozov	N
24	Electric Drive Systems and Operation	Valery Vodovozov	N
25	Introduction to Electronic Engineering	Valery Vodovozov	N
26	Introduction to Power Electronics	Valery Vodovozov	N
27	Electrical Power	W. J. R. H. Pooler	N
28	Concepts in Electric Circuits	Wasif Naeem	N
29	Introduction to Digital Signal and System Analysis	Weijing Wang	**Y**
30	Artificial Intelligence-Agent Behavior	William John Teahan	N
31	Artificial Intelligence-Agents and Environments	William John Teahan	N
32	Advanced Granulation Theory at Particle Level	Peter Dybdahl Hede	N
33	A first Course in Fluid Mechanics for Engineers	Buddhi N. Hewakandamby	N
34	A first Course on Aerodynamics	Amab Roy	N
35	Atmospheric Pollution	J.C. Jones	N
36	Automation and Robotics	Miltiadis A. Boboulos	N
37	A Wet Look At Climate Change	Peter Moir	N
38	Bioethanol: Science and technology of fuel alcohol	Graeme M. Walker	N
39	CAD-CAM & Rapid prototyping Application Evaluation	Miltiadis A. Boboulos	N
40	Chemical Thermodynamics	Leo Lue	N
41	Chemistry for Chemical Engineers	Ashleigh J. Fletcher	N
42	Introductory Maths for Chemists	J.E. Parker	N
43	Intermediate Maths for Chemists	J.E. Parker	N
44	Compuatational Fluid Dynamics	Abdulnaser Sayma	N
45	Drilling Fluid Engineering	Pal Skalle	N
46	Electrically Driven Membrane Processes	Soren Prip Beier	N
47	Engineering Fluid Mechanics	Tarik Al-Shemmeri	N
48	Fluid Bed Particle Processing	Peter Dybdahl Hede	N
49	Food Processing	Carl J. Schaschke	N
50	Fundamentals of Chemistry	Romain Elsair	N
51	Fundamentals of Hydrogen Safety Engineering I	Vladimir Molkov	N
52	Fundamentals of Hydrogen Safety Engineerring 2	Vladimir Molkov	N
53	Fundamentals of Reaction Engineering- Examples	Rafael Kandiyoti	N
54	Hydrocarbons	J. C. Jones	N
55	Hydrodynamic Modelling and Granular Dynamics	Peter Dybdahl Hede	N
56	Introduction to Cancer Biology	Momna Hejmadi	N
57	Introduction to Clinical Biochemistry	Graham Basten	N
58	Introduction to Inorganic Chemistry	Peter G. Nelson	N
59	Introduction tPoylmer Science and Technology	Mustafa Akay	N
60	Introduction to Scientific Research Projects	Graham Basten	N
61	Basic Concepts in Trubomachinery	Grant Ingram	N
62	Micro- and Nano- Transport of Biomolecules	David Bakewell	N
63	Minerals and Rocks	J. Richard Wilson	N
64	Modelling Batch Systems Using Population Balances	Peter Dybdahl Hede	N
65	Molecular Conformations	Christopher Wood	N
66	Essentials of Nanotechnology	Jeremy Ramsden	N
67	Pharmacokinetics	Philip Rowe	N
68	Pressure Control During Oil Well Drilling	Pal Skalle	N
69	Pressure driven Membrane Processes	Soren Prip Beier	N
70	Theory of waves in materials	Jeremiah Rushchitsky	N
71	Java: Classes In Java Applications	David Etheridge	N
72	Java:Graphical User Interfaces	David Etheridge	N
TOTAL	72 books	51 authors	6 PAN books

APPENDIX C FULL FIGURES OF DESCRIPTIVE ANALYSIS

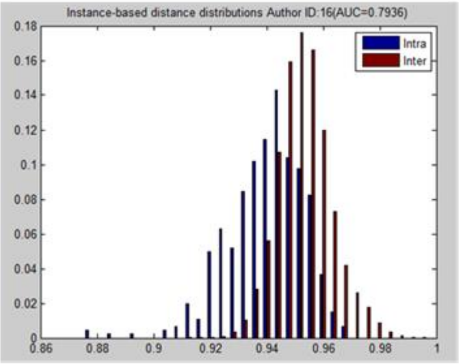


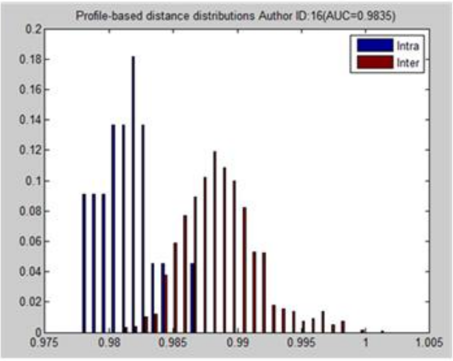

Instance-based Distance
Profile-based Distance
Author 13
Instance-based distance distributions Author ID:13(AUC=0.9916)
Profile-based distance distributions Author ID:13(AUC=0.9998)
Author 14
Instance-based distance distributions Author ID:14(AUC=0.9954)
Profile-based distance distributions Author ID:14(AUC=1.0000)
Author 15
Instance-based distance distributions Author ID:15(AUC=0.9735)
Profile-based distance distributions Author ID:15(AUC=0.9991)
Intra
Inter

Instance-based Distance

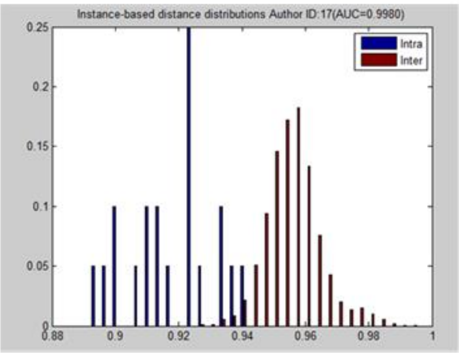
Profile-based Distance

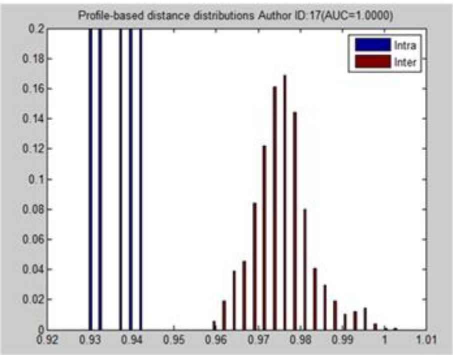
Author 16



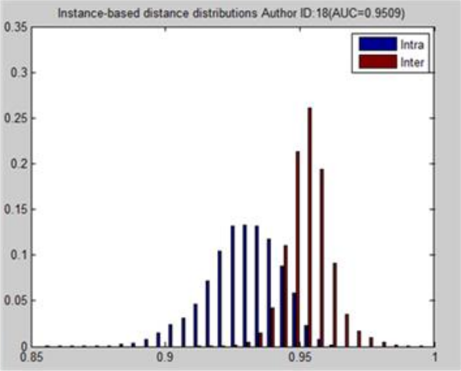


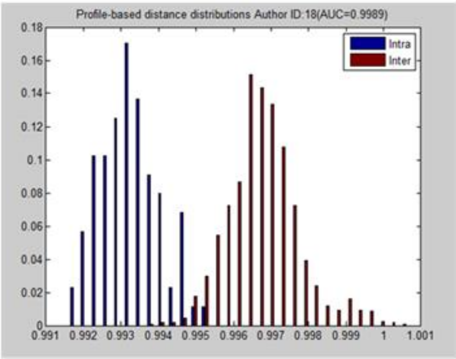
Author 17

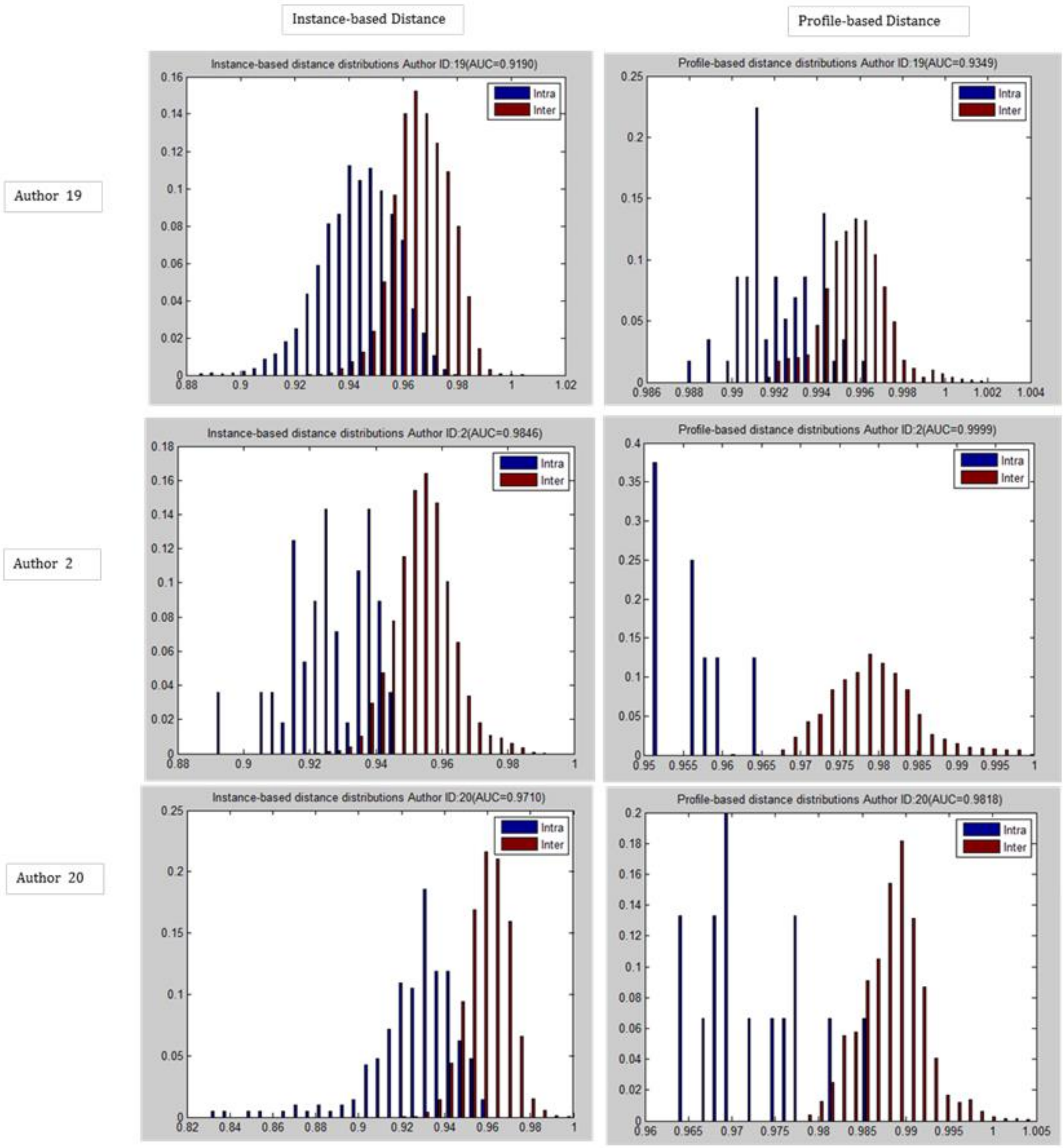


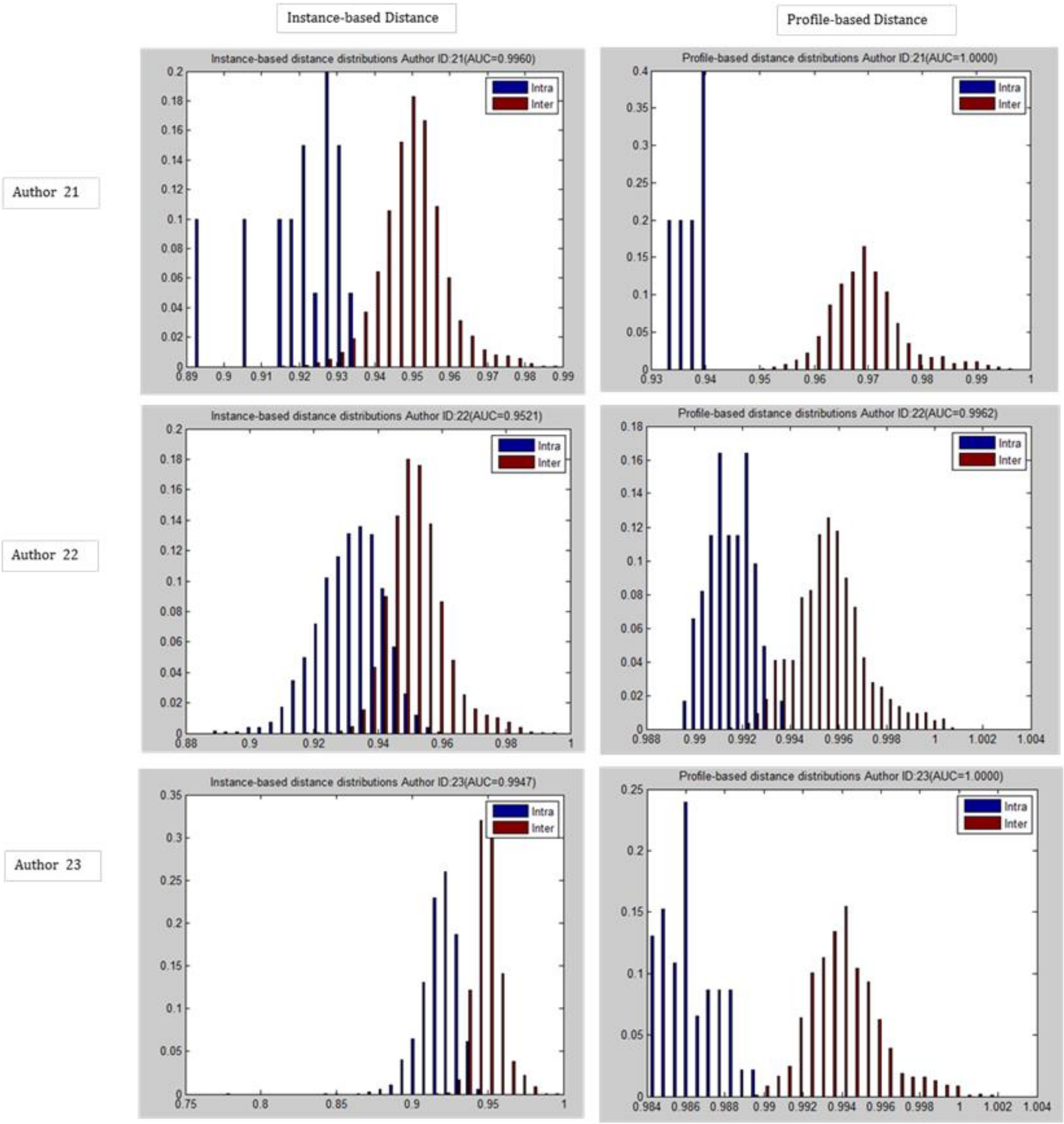


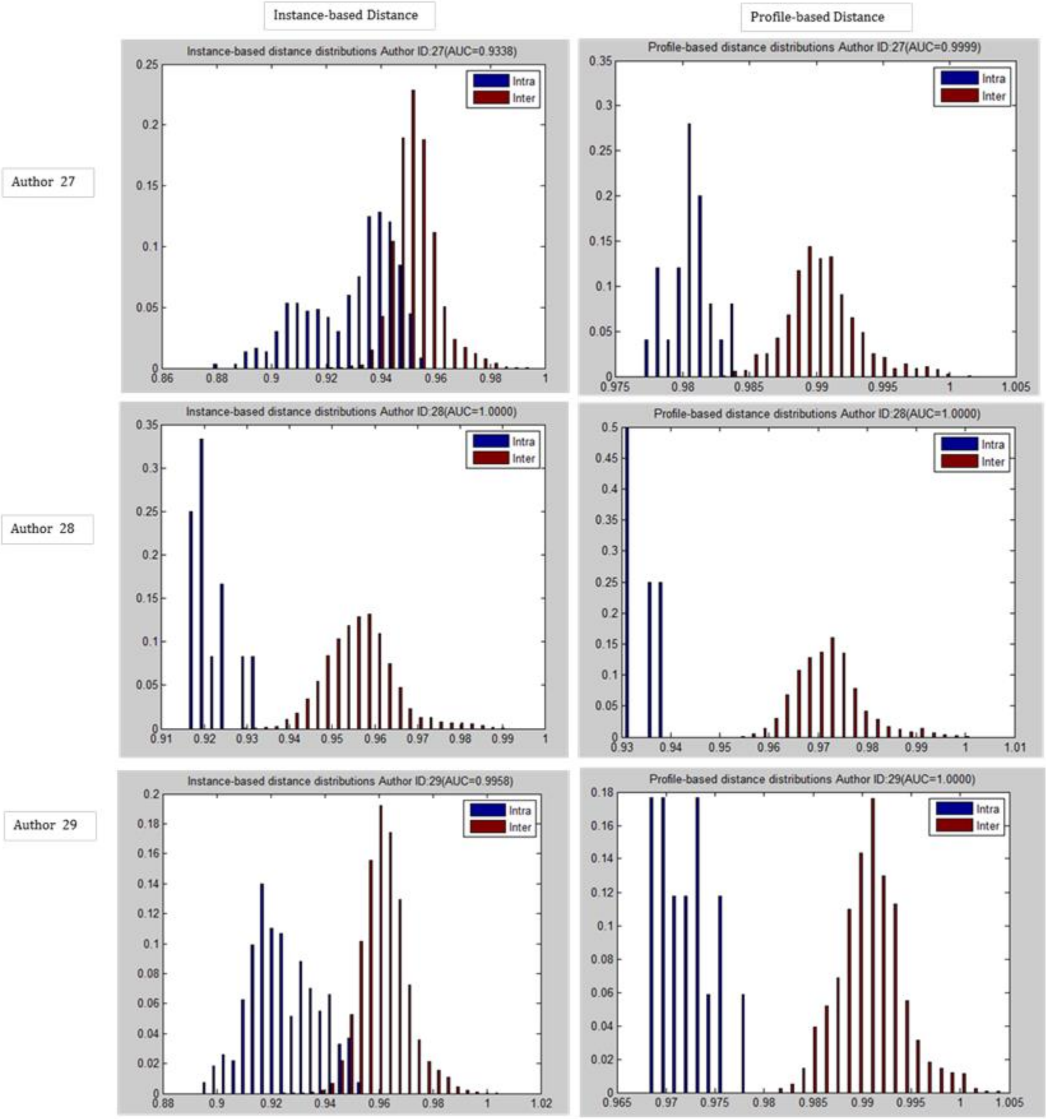
Author 18






Instance-based Distance
Profile-based Distance
Author 19
Author 2
Author 20
Instance-based distance distributions Author ID:19(AUC=0.9190)
Profile-based distance distributions Author ID:19(AUC=0.9349)
Instance-based distance distributions Author ID:2(AUC=0.9846)
Profile-based distance distributions Author ID:2(AUC=0.9999)
Instance-based distance distributions Author ID:20(AUC=0.9710)
Profile-based distance distributions Author ID:20(AUC=0.9818)
Intra
Inter
0 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16
0.88 0.9 0.92 0.94 0.96 0.98 1 1.02
0 0.05 0.1 0.15 0.2 0.25
0.986 0.988 0.99 0.992 0.994 0.996 0.998 1 1.002 1.004
0 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16 0.18
0.88 0.9 0.92 0.94 0.96 0.98 1
0 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4
0.95 0.955 0.96 0.965 0.97 0.975 0.98 0.985 0.99 0.995 1
0 0.05 0.1 0.15 0.2 0.25
0.82 0.84 0.86 0.88 0.9 0.92 0.94 0.96 0.98 1
0 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16 0.18 0.2
0.96 0.965 0.97 0.975 0.98 0.985 0.99 0.995 1 1.005


Instance-based Distance
Profile-based Distance
Author 21
Author 22
Author 23
Instance-based distance distributions Author ID:21(AUC=0.9960)
Profile-based distance distributions Author ID:21(AUC=1.0000)
Instance-based distance distributions Author ID:22(AUC=0.9521)
Profile-based distance distributions Author ID:22(AUC=0.9962)
Instance-based distance distributions Author ID:23(AUC=0.9947)
Profile-based distance distributions Author ID:23(AUC=1.0000)
Intra
Inter
0 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16 0.18 0.2
0.89 0.9 0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99
0 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4
0.93 0.94 0.95 0.96 0.97 0.98 0.99 1
0.88 0.9 0.92 0.94 0.96 0.98 1
0 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16 0.18
0.988 0.99 0.992 0.994 0.996 0.998 1 1.002 1.004
0 0.05 0.1 0.15 0.2 0.25 0.3 0.35
0.75 0.8 0.85 0.9 0.95 1
0 0.05 0.1 0.15 0.2 0.25
0.984 0.986 0.988 0.99 0.992 0.994 0.996 0.998 1 1.002 1.004


Instance-based Distance
Profile-based Distance
Author 27
Author 28
Author 29
Instance-based distance distributions Author ID:27(AUC=0.9338)
Profile-based distance distributions Author ID:27(AUC=0.9999)
Instance-based distance distributions Author ID:28(AUC=1.0000)
Profile-based distance distributions Author ID:28(AUC=1.0000)
Instance-based distance distributions Author ID:29(AUC=0.9958)
Profile-based distance distributions Author ID:29(AUC=1.0000)
Intra
Inter
0.86 0.88 0.9 0.92 0.94 0.96 0.98 1
0.975 0.98 0.985 0.99 0.995 1 1.005
0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99 1
0.93 0.94 0.95 0.96 0.97 0.98 0.99 1 1.01
0.88 0.9 0.92 0.94 0.96 0.98 1 1.02
0.965 0.97 0.975 0.98 0.985 0.99 0.995 1 1.005

APPENDIX D AUC OF THE EMAIL CORPUS

NO.	AUC (Instance)	AUC (Profile)
1	0.6384	0.9541
2	0.3510	0.9143
3	0.5881	0.9060
4	0.5245	0.9461
5	0.5956	0.9653
6	0.9099	0.9797
7	0.4229	0.8334
8	0.5207	0.9744
9	0.9943	1.0000
10	0.5640	0.9662
11	0.5109	0.8795
12	0.6252	0.9485
13	0.5974	0.9685
14	0.6223	0.9595
15	0.4403	0.9014
16	0.5129	0.9448
17	0.4867	0.8555
18	0.6086	0.9770
19	0.5446	0.9407
20	0.4281	0.8724
21	0.6569	0.9491
22	0.4545	0.8622
23	0.0643	1.0000
24	0.6242	0.9701
25	0.5886	0.9968
26	0.3962	0.9666
27	0.5890	0.9579
28	0.4759	0.9641
29	0.5419	0.9833
30	0.4407	0.8490
31	0.5405	0.9594
32	0.6122	0.9687
33	0.6027	0.6919
Average	0.5477	0.9335