Authorship Verification with Skip N-grams and Termsets

Chao Chen B.Eng.

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Intelligent Systems)

Supervisor: Carl Vogel, Erwan Moreau

September 2020

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Chao Chen

September 7, 2020

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Chao Chen

September 7, 2020

Acknowledgments

Thanks a million to Prof. Carl Vogel, for his tolerance and guidance.

Also many thanks to Dr. Erwan Moreau, though I implement the code by my own, some of the ideas are from his previous work.

Thanks to my brother Xia and his family, they always support me when things happen.

We all have been through a special time under COVID-19 pandemic.

Finally, to my mother and father, hope I could receive their praise.

Chao Chen

University of Dublin, Trinity College
September 2020

Authorship Verification with Skip N-grams and Termsets

Chao Chen, Master of Science in Computer Science

University of Dublin, Trinity College, 2020

Supervisor: Carl Vogel, Erwan Moreau

The aim of this work is to go through a complete process to solve Authorship Verification problems. A set of problems are given in the Authorship Verification task, in each problems, several known texts and one unknown texts are present. The aim is to judge whether these texts are written by the same author.

The experiments focus the difference between two kinds of features: skip n-grams and termsets. A skip n-gram is a special n-gram that can take whatever types of token into account in some of its positions. Weighted termsets are terms without spacial relations, there weighting depends on the occurrence across the problem texts, by whether they appear simultaneously or alone. The termset weighting method is adapted to Authorship Verification base on small amount of texts.

This work follows the rules and uses the datasets from PAN at CLEF shared tasks in 2013, 2014 and 2015, where classifications are implemented with corpora from different genres with different topics. Classifications with SVMs, random decision trees are performed. The coefficients of linear kernel SVMs are examined to identify the influences of features during classification.

A modified Impostor method and an adaption of Universum Inference are also selected as option strategies. The modified Impostor method details the scoring for each repetition and adds a filter for impostors with better similarities before scoring. The Universum Inference is simplified to solve the problems with small sets of texts.

The genre may have impact on the skip n-gram and termset performance. Skip n-gram features seem to be more effective than the termset features in a corpus close to oral form.

Contents

Abbreviations

AUC	Area Under Curve
BOW	Bag-of-Words
CDM	Compression Dissimilarity Measure
CLEF	Conference and Labs of the Evaluation Forum
GI	General Impostors
HTML	Hyper Text Markup Language
idf	inverse document frequency
kNN	k-Nearest-Neighbor
LESS	Lowest Error in a Sparse Subspace
POS	Part-of-Speech
RBF	Radial Basis Function
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
tf	term frequency
TTR	Token-Type Ratio

List of Tables

List of Figures

Chapter 1

Introduction

Authorship Verification is a process to judge whether two set of text are written by the same author. An algorithm will be used to automatically extract the features or linguistic styles of articles or emails, and then be compared to attribute the authorship or analyze the situation or feeling of the author.

Authorship Verification is a subset of Authorship Identification problems, in which we are asked to analyze the characteristics of the author to help solving problems. Existing Authorship Attribution and Authorship Clustering problems can all be deconstructed into Authorship Verification problems (Koppel et al., 2012).

Authorship Verification can play a critical role in judicial or journalistic cases to find the authentic authors of anonymous or disputed documents. Techniques derived from authorship verification can be also used to detect plagiarism (Stamatatos, 2009) and to classify text sentiment (Malmasi et al., 2016; HaCohen-kerner et al., 2017). Anonymous emails can also be grouped by their authors using Authorship Verification techniques. Authorship Verification is becoming increasingly important nowadays, with the exponential expansion the information technology. The text contents, including social medias, emails, web pages, forums, etc. are expanding at a enormous speed, which make it almost impossible for human to keep a track on the authorship problems when needed. On the other hand, with the large number of digital traits left on the internet, and more computational power available nowadays, the feasibility for authorship verification is much better than in the past.

Typical characteristics of a text are the counts of various n-grams or Bag-of-Words

model (BOW). In this work, I am going to focus on two specific types of characteristics - skip n-grams and weighted termsets. While n-gram is a contiguous sequence of "n" items from the text, skip n-grams have a number of gaps included, and the weighted termsets are calculated without ordering and positional relations between two or more terms.

1.1 Scope and Aims

For this work aims to identify the improvements and do close observations on the classification process with various strategies, general classifiers like Support Vector Machines (SVMs) and Random Decision Trees are used and also the Impostor method and Universum Inference are selected as optional strategies.

One idea to observe the "importance" of one characteristic is get the parameter or coefficient of it, so linear classifiers will be examined in detail. Neural networks are not considered in this work for the difficulty to examine detailed characteristics.

The problem settings are stick to the definition from PAN at CLEF Webis group (2020) shared tasks. There are 3 shared tasks on Authorship Verification problems and results published before 2020 (Stamatatos et al., 2013, 2014, 2015). The datasets contains texts across topics, genres and even languages. Due to limited knowledge about other languages, here we just focus on Authorship Verification problems with only English texts.

There are several goals and hypotheses:

- **G1.** Go through a complete Authorship Verification process using selected methods.
- **H1.** There is an improvement in performance of classification using general classifiers, the Impostor method and Universum Inference after introducing skip n-grams.
  - Features related to skip n-grams has a higher weight of coefficient of a linear classifier (SVM).
- **H2.** There is an improvement in performance of classification using general classifiers, the Impostor method and Universum Inference after introducing termsets.

– Features related to termsets has a higher weight of coefficient of a linear classifier (SVM).

1.2 Structure of Thesis

This thesis attempts to identify the changes before and after introducing skip n-grams and termsets in 3 strategies used in Authorship Verification problems: general classifiers, the Impostor method and the Universum Inference method.

Chapter 2 introduces background and literature reviews on the problem, methods, and previous works on Authorship Verification problems.

Chapter 3 explains the concrete methods and implementations of the experiments in detail.

Chapter 4 presents the experiments on 3 selected datasets and their results, with a overall observation.

Chapter 5 draws the conclusion of this work, as well as several observations gained in each of the experiments. Future works are also brought during the discussion.

Chapter 2

Background

This chapter contains the background of my research and a literature review on the previous works related to the project. I will explain the problems I am trying to solve. Several observations, or features of texts, are introduced. Then I will go through common strategies used in the Authorship Verification, mainly about applying classifiers such as Support Vector Machines (SVMs), and statistic methods like the Impostor method and Universum Inference method. Brief summaries of the PAN at CLEF Authorship Verification shared tasks for year 2013, 2014 and 2015 will be provided, and the outstanding candidates related to the method used in this project will be reviewed

To avoid confusion, I am calling features obtained from one text "observations", while numbers calculated from "observations" and prepared for training process "features". For example, the count or tf-idf of a word bi-gram is an "observation", a number representing the distance on one kind of bi-gram of known and unknown texts is a "feature".

2.1 Authorship Verification

The main goal of Authorship Verification is to determine if any two documents are written by the same author (Koppel and Winter, 2014). Authorship Verification is commonly used in business or legal scenarios when identifying the authorship of a document are critical, such as for a pseudonymous or anonymous report article or a blackmail.

Existing problem setups such as Authorship Attribution or Authorship Clustering problems can be deconstructed into Authorship Verification problems (Koppel et al., 2012). The process to group texts by stylometric features of a same author are Authorship Clustering (Iqbal et al., 2010). These tasks can all be concluded as Authorship Identification tasks. The science of identifying characteristics of authors from a dataset written by the authors (Juola, 2006) is called the Authorship Attribution problem.

Cross-domain Authorship Verification is then brought out, where there are loose or no constraints to the form of the texts, in which more research effort is needed. Authorship Verification tasks can be more challenging when they are applied to cross-domain conditions, which means the texts of the known and unknown datasets are from different domains (e.g. genre of texts, themes). Some of my dataset selected will be cross-topic, but still in the same genre.

2.2 Stylometric Observations

Characteristics of text can be expressed as a vector. Each element in the vector is a number related to a specific feature or pattern, the number is often operated from the count.

The counted feature might be a word as part of the bag-of-words (BOW) model, or it can be an n-gram. an n-gram is a contiguous sequence of "n" items from a given sample of text or speech. The items can be can be phonemes, syllables, letters, words or base pairs according to the implementations.

Stamatatos (2009) and El and Kassou (2014) did a detailed survey on the stylometric features and automated approaches to attributing authorship, in which features are classified into lexical, character, syntactic, semantic and application-specific features. What I am focusing is the syntactic, lexical and character features, both concepts I am going to introduce are not often used in authorship identification tasks.

This work aims to provide a observation on the performance improvement after introducing skip n-grams and termsets.

2.2.1 Skip N-Grams

Besides various kinds of n-grams, I can add information of "gaps" between word, tokens, etc., to form skip n-grams (Guthrie et al., 2006). For example, the sentence "Tanker likes to play football" has limited an array of **skip bi-grams** within 2-skip as:

tanker, likes;
likes, to;
to, play;
play, football;
tanker, ⟨ skip ⟩, to;
likes, ⟨ skip ⟩, play;
to, ⟨ skip ⟩, football;
tanker, ⟨ skip ⟩, ⟨ skip ⟩, play;
likes, ⟨ skip ⟩, ⟨ skip ⟩, football;
tanker, ⟨ skip ⟩, to, ⟨ skip ⟩, football.

We can see that a skip n-gram counts irrelevant or unselected words as gaps. Intuitively thinking, I expect skip n-grams to capture syntactic structures omitting some decorative words according to Part-of-Speech in English such as adverbs and phrases.

Pokou et al. (2016) used part-of-speech skip-grams with an in-house top-k sequential pattern mining algorithm for the task of authorship attribution. They studied on a dataset of 30 large texts from 10 authors, achieving an average accuracy between 85.8%, 92.6%, and 93.34% for selecting 50, 100, and 250 skip n-grams with $maxgap = 1$ (only one continuous skip sequence appear in a skip n-gram) as features.

Malmasi et al. (2016) tried to predict the severity of user posts in a mental health forum. They employed a meta-classifier which uses a set of of base classifiers, mainly SVMs, constructed from lexical, syntactic and metadata features including 1, 2 and 3-skip word bi-grams.

A similar work (HaCohen-kerner et al., 2017) also implemented a classifier using SVM on the stance of the twitter user towards several topics, whether they were positive, negative, or neutral. A large number if skip n-gram features are sampled with a number both 15,000 for character and word skip n-grams.

2.2.2 Termsets

Termset, also known as itermset, association features, term associations (Meretakis and Wüthrich, 1999; Zhang Yang et al., 2003; Tesar et al., 2006), or word co-occurrence (Figueiredo et al., 2011). It is a special kind of bag of words with multiple frequent words occur in the text as one feature. Intuitively I have the idea that termsets represents the topic of the text, since frequently associated words or stemmed terms are mined and selected by a threshold.

In stead of calculating values with isolated termsets, we can consider termsets to have negative effect to each other. Discrimination can be applied when one termset often occurs in a text while some others do not.

A study by Badawi and Altınçay (2014) introduces a new method for 2-termset selection and weighting by employing statistics of the joint occurrence statistics of pairs of terms. Instead of binary weighting or weighting just based on only the combination, each termset is evaluated by considering the individual term, or simultaneous occurrences of the terms within the termset, which means rather than focusing only on whether they both occur or not, the method also consider the cases where only one of the selected terms appears but bot the other. 500 termsets were collected using χ^2 for selection and SVM for classification on three large text datasets. Improvements were achieved when author added termsets with Bag-of-Words (BOW) model.

The termset weight method are introduced here:

For term t_i, t_j:

- A: the number of positive documents including both t_i and t_j.
- C: the number of negative documents including both t_i and t_j.
- P: the number of positive documents including t_i but not t_j.
- Q: the number of negative documents including t_i but not t_j.
- R: the number of positive documents including no t_i but t_j.
- S: the number of negative documents including no t_i but t_j.

Then we have the relevance frequency RF:

$$RF(\{t_i, t_j\}) = \begin{cases} log_2(2 + \frac{A}{max(C,1)}) & \text{both } t_i \text{ and } t_j \text{ occur} \\ log_2(2 + \frac{P}{max(Q,1)}) & t_i \text{ occurs but not } t_j \\ log_2(2 + \frac{R}{max(S,1)}) & t_j \text{ occurs but not } t_i \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

And the weighted frequency of $wf(\{t_i, t_j\})$:

$$wf(\{t_i, t_j\}) = (tf_i + tf_j) \times RF(t_i, t_j) \tag{2.2}$$

2.3 Strategies

Two optional types of classification strategies are used in the experiments of this work besides general classifiers.

Machine learning methods based on neural networks like Recurrent Neural Networks (RNNs) are not considered in this work. Since my aim is to compare the performance with and without specific features, and have a closer look on the linear outcome of the classifiers, the internal process of neural networks is not readable comparing with linear classifiers such as linear kernel SVMs.

2.3.1 The Impostor Method

The Impostor method (Koppel and Winter, 2014) uses a set of external documents from other author (not the authors under investigation) to build a random set.

Tests are carried out to judge whether the candidate author or the generated impostors are closer to the disputed text.

- Generate a set of impostors $Y_1, ..., Y_m$
- Compute $score_X(Y)$ = the number of choices of feature sets (out of 100) for which $sim(X, Y) > sim(X, Y_i)$
- Repeat the above with imposters $Y_1, ..., Y_m$ and compute $score_Y(X)$ in analogous manner.

- If $average(score_X(Y), score_Y(X))$ is greater than a threshold Δ^*, assign $\langle X, Y \rangle$ to "same-author".

The training process is to find the best Δ^*.

n order to adapt the original impostors method to the settings of Authorship Verification problems from PAN at CLEF, General Impostors (GI) method is introduced in the consideration identifying multiple documents of one candidate author to identify (Seidman, 2013), the algorithm is described as Algorithm 1:

Algorithm 1: General Impostors method

input : $\langle K, U \rangle$: A pair of documents, Known and Unknown.
S: A set of impostors.
$rate\%$, k, n, Δ^*.

output: $\langle same - author \rangle$ or $\langle diff - author \rangle$

Set $Score = 0$;

for *i=1* **to** k **do**

  Randomly select $rate\%$ of the features from the full feature pool;

  Randomly select n impostors $I_1, ..., I_n$ from S;

  $Score = Score + 1/k$ **if** $sim(K, U) * sim(U, K) > sim(K, I_j) * sim(U, I_j)$ **for each** $j \in 1, ..., n$;

end

Return $\langle same - author \rangle$ **if** $Score > \Delta^*$; **else** $\langle diff - author \rangle$

Potha and Stamatatos (2017) introduced an improved Impostor method by adding a pre-selecting impostors process before scoring. Also the scores are detailed with similarity rankings. This proposed algorithm proposed outperforms the original GI using only character 5-gram. A general algorithm is described in Algorithm 3.

Similarities used in the presented previous works (Koppel and Winter, 2014; Seidman, 2013; Potha and Stamatatos, 2017) are cosine similarity (Eq.3.2) and MinMax similarity (Eq.3.3).

Both Seidman (2013) and Potha and Stamatatos (2017) use search engines to produce web impostor corpus with selected small sets of words (by top tf-idf or ran-

domly selected from known texts), top web pages are downloaded and HTML tags are stripped. Then the impostors are chopped to a similar length to the problem texts.

It should be noticed that the genre of the impostors have impact on the performance of classification, if the texts are cross-genre, the performance will downgrade.

2.3.2 Universum Inference

Universum Inference, originally raised by Vogel et al. (2009), focuses on the homogeneity of play scripts by calculating the homogeneity of partial texts from the original problem dataset. A large corpus, which is originally tagged as different "categories" respect to the author of the texts, is split into small chunks. Each chunk of the original category is compared with chunks of texts from other categories as well as text chunks from the same category. Similarities are measured using χ^2 tests and the homogeneity is measured using Mann-Whitney rank ordering statistic through Bernoulli Schema. Only the categorization or classification with enough homogeneity measured will be recorded.

Moreau et al. (2015) adapts Universum Inference to fit the setting of Authorship Verification tasks from PAN. The general idea is to measure the difficulty to identify texts from different categories:

Algorithm 2: Universum Inference Adaption (Moreau et al., 2015)

input : $\langle K, U\rangle$: A pair of documents, Known and Unknown.
n.

output: $\langle same - author\rangle$ or $\langle diff - author\rangle$

for *i=1* **to** k **do**

Split K and U randomly with similar length $\{K_1, K_2, K_3\}, \{U_1, U_2, U_3\}$;

K_3 and U_3 are split again into $\{K_3', K_3''\}, \{U_3', U_3''\}$;

Texts are categorized into 3 classes: $C_K = \{K_1, K_2\}$, $C_U = \{U_1, U_2\}$, $C_{mixed} = \{K_3' \cap U_3', K_3'' \cap U_3''\}$;

Calculate $Score_i$: measure the confusion of the similarities between the 6 texts from 3 categories. Higher the confusion, higher the score;

Record $Score_i$;

end

$Score = aggregate(Score_i, ..., Score_n)$;

Return $\langle same - author\rangle$ **if** $Score > \Delta^*$; **else** $\langle diff - author\rangle$

Several approaches are used to measure the confusion $Score_i$. The first idea is to rank the similarities measured between the 6 texts, texts in the same category should have the highest similarity, while texts from C_K and C_U should have the lowest similarities. Texts from C_{mixed} will blur the difference when comparing with K or U, since texts from C_{mixed} are the combination of K and U.

The adaption of Universum Inference is reported to perform well locally in Moreau et al. (2015), but the overall performance is unstable.

2.4 PAN at CLEF

There are shared tasks available related to Authorship Verification. An overview on the tasks of PAN at CLEF (Webis group, 2020) related to Authorship Verification is presented in this section.

It should be noticed that I only focus on English documents rather than Dutch, Spanish or other languages, so when I am viewing the literature with rankings high in

English categories rather than overall scores in the overviews (Stamatatos et al., 2013, 2014, 2015).

There are sets of problems for each language, and a problem contains at least one known document from the known author, and exactly one unknown document to judge whether it is from the same author.

A few shared tasks are shown in Table 2.1. The methods used that achieved high rankings are also listed in the table.

Table 2.1: Shared tasks of PAN at CLEF on Authorship Verification

Workshop	**Task**	**Features Used**	**Description**
PAN 2013	Verification (Stamatatos et al., 2013; Seidman, 2013)	n-gram / character n-gram / function words	Impostor & General Impostors with MinMax similarity
PAN 2014	Verification (Stamatatos et al., 2014; Frery et al., 2014)	n-gram / character n-gram / phrases / vocabulary diversity / punctuation	mostly tf-idf using correlation coefficient
PAN 2015	Verification (Stamatatos et al., 2015; Bagnall, 2015)	original text pre-processing by character mapping	RNN

2.4.1 PAN 2013

PAN 2013 (Stamatatos et al., 2013) is the first problem of PAN at CLEF using the Authorship Verification settings. The corpora cover 3 language genres: English, Greek and Spanish. The corpus is collected from textbooks on computer science or related subjects published in an on-line repository. Since the genre and topic is limited, the universe of discourse is relatively controlled, also this kind of corpora is relatively

unstudied compared to newspaper articles or fictions. Top 3 solutions in English genre are listed in Table 2.2.

Table 2.2: Top 3 of PAN 2013 - English Genre

Solution	F_1	*Precision*	*Recall*
Seidman (2013)	0.800	0.800	0.800
Veenman and Li (2013)	0.800	0.800	0.800
Layton et al. (2013)	0.667	0.667	0.667

Seidman (2013) ranked top overall and in the English part. GI is proposed in this work to adapt the Impostor methods to Authorship Verification settings. Function words, word n-grams and character n-grams are evaluated with binary, numeric and tf-idf methods.

Veenman and Li (2013) also ranked top in the English genre using Compression Dissimilarity Measure (CDM) (Keogh et al., 2004). Compression distances are measured and Sparse Subspace (LESS) classifier (Veenman and Tax, 2005) is used to category the documents with minimum error rate.

Layton et al. (2013) applies Local N-grams (Kešelj et al., 2003) for Authorship Verification. Character n-grams that occur simultaneously in both known and unknown texts in a pair are taken into consideration and the distance of the pair is calculated. A best threshold of distance is found during training process.

2.4.2 PAN 2014

PAN 2014 Stamatatos et al. (2014) covers Dutch, English, Greek, and Spanish contents. The English part contains 2 subsets of problems: essays and novels.

The English essays corpus is derived from the Uppsala Student English corpus Axelsson (2000), the essays come from English-as-second-language students of university-level full-time education in electronic forms.

The novels corpus consists of speculative and horror fictions known as the "Cthulhu Mythos". This series of novels are originally written by American author H.P. Lovecraft and are extended by other writers as fan-fictions. The novels section is naively considered to be a narrower universe comparing the essays, since the topic is limited and the vocabulary is unusual (e.g., unpronounceable proper names of these cosmic horrors such as "Cthulhu", "Nyarlathotep", "Lloigor", "Tsathoggua", and "Shub-Niggurath").

The non-response measurement $c@1$ Peñas and Rodrigo (2011) is introduced since PAN 2014. The performance measures are mainly using $AUC * c@1$. The results are listed in the following table:

Table 2.3: Top 3 of PAN 2014 - English Genre

Solution	$AUC * c@1$	AUC	$c@1$
English Essays			
Frery et al. (2014)	0.512	0.723	0.710
Satyam et al. (2014)	0.459	0.699	0.657
Moreau et al. (2014)	0.372	0.620	0.600
English Novels			
Modaresi and Gross (2014)	0.508	0.711	0.715
Zamani et al. (2014)	0.476	0.733	0.650
Khonji and Iraqi (2014)	0.458	0.750	0.610

Frery et al. (2014) collects word n-grams, character n-grams, phrases, vocabulary diversities and punctuation as features, cosine similarities and correlation coefficient are calculated, Classification and Regression Trees are used as classifiers.

Satyam et al. (2014) generate character n-grams and generate various schemes like term-frequency, log-term-frequency or binary term-frequency for local weighting, Entropy and idf are used as the global weighting methods. After confidence measure, cosine similarities and Jaccard similarities are calculated, and a threshold is deter-

mined after training.

Moreau et al. (2014) implement a SVM classifier with large set of parameters. Words, characters, POS, stopwords and token length classes n-grams, TTRs are collected and filtered from texts based on a set of configurations. Then features are calculated based on 4 aspects: consistency, divergence, confidence and distance. The observations generation and filtering of my work is mainly based on this previous work, details will be explained further in the next chapter).

The approach from Modaresi and Gross (2014) uses fuzzy clustering with basic lexical feature extraction. Average sentence length, punctuation marks usage, space before and after comma are considered to generate vectors representing the documents. Personality traits regarding Big Five Personality Traits Digman (1990) are also collected from each texts. Fuzzy C-Means Bezdek et al. (1984) algorithm is used for clustering.

Zamani et al. (2014) store sets of features from problem texts, the feature sets consist of probabilistic distribution of stopword, punctuation, n-grams, sentence length, paragraph length, POS tags and word length. Distances are calculated using several divergence method and kNN algorithm is implemented for classification.

Khonji and Iraqi (2014) adapt GI method (Seidman, 2013) by a more detailed scoring method when comparing impostors with problem texts. Various combinations of n-grams on characters, words, function words, word shapes (upper-cases or lower-cases), POS tags and POS-tagged words.

Stamatatos et al. (2014) implement a meta-classifier combining all the submitted systems by calculating the average of probability scores of all submitted results achieves the best overall performance. Later work by Moreau et al. (2015) try to implement the meta classifier based on this observation using genetic approach. For each known texts, 10 impostors with top similarities are collected by searching random words from the known texts in a search engine.

2.4.3 PAN 2015

Authorship Verification task in PAN 2015 Stamatatos et al. (2015) contains 4 language corpora: Dutch, English, Greek and Spanish. The English corpus comes from lines the actors speak on stage in play scripts cross different topics. The count of English test problem set is extra large comparing its training set and also with other languages.

Table 2.4: Top 3 of PAN 2015 - English Genre (*:data lack of precision)

Solution	$AUC * c@1$	AUC	$c@1$
Bagnall (2015)	0.614	0.81*	0.76*
Castro et al. (2015)	0.520	0.750	0.694
Gutierrez et al. (2015)	0.513	0.74*	0.69*

Castro et al. (2015) collect character, lexical and syntactic level of features from the problem texts, including character n-grams, character prefixes and suffixes, word n-grams, POS n-grams and lemma n-grams. Similarities are measured using MinMax similarity, pairwise similarities are measured and judged to identify whether the unknown file is written from the author of known files.

Gutierrez et al. (2015) implemented a classification process based on GI method by introducing homotopy into the similarity measurement between problem texts and impostors. A vector is calculated by L1-homotopy algorithm (Asif and Romberg, 2013) to reconstruct features selected from the known texts with their impostors. The reconstructed feature vector can be seemed to be the aggregation of the known and impostors, so the unknown texts are accessed universally.

2.4.4 Conclusion

A classic process of Authorship Verification contains following steps: observation extraction, feature generation, training process. Character n-grams seems to be one of the popular characteristics of texts among the solutions with its tf-idf form, and cosine similarity are often used as the similarity or distance measure. No weighted termset features and few skip n-grams are applied int the solutions reviewed. Since the datasets are relatively small comparing other authorship identification problem settings, methods using external corpora such as the Impostor methods are applied to extend the dataset. There is a trend to apply neural networks (especially RNNs) to Authorship

Verification problem, since it is vague to represent the coefficients of parameters or features, neural networks are not included in the experiments of this work.

Chapter 3

Method

The method I am using in this work are mainly picked and from works of Moreau et al. (2014, 2015) with several changes. Moreau et al. (2014, 2015) implement a complex genetic process to select meta classifiers with best performance using long CPU hours. Since it's not our goal to exceed the performance of the meta classifiers, the process solving Authorship Verification problem is simplified based on both previous works and aims of this work.

3.1 Equations

Besides equations introduced in Chapter 2, there are also equations and expression I use in this work. Some equations are listed but not used directly in the work, but give inspirations to form new formulas.

3.1.1 Term Frequency

Let $n_{i,j}$ be the count of the term i in text j, then we can get the term frequency (tf) of the term i in text j:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \approx \frac{n_{i,j}}{token\_count} \tag{3.1}$$

3.1.2 Similarity

Cosine Similarity

We calculate cosine similarity by Eq.3.2:

$$sim(X, Y) = cosine(X, Y) = \frac{X \cdot Y}{||X|| \times ||Y||} \tag{3.2}$$

MinMax Similarity

We calculate MinMax similarity by Eq.3.3:

$$sim(X, Y) = minmax(X, Y) = \frac{\sum_{i=1}^{n} min(x_i, y_i)}{\sum_{i=1}^{n} max(x_i, y_i)} \tag{3.3}$$

Jaccard Similarity Coefficient

Jaccard similarity coefficient, or Jaccard index, can be calculated by Eq.3.4:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{3.4}$$

Where A, B are the collections of observations from two texts.

3.1.3 Evaluation Metrics

This section introduces several metrics normally used in Authorship Verification tasks. A list of statistics that can be gathered during the implementations is shown as below:

- n_{ac}: number of questions for which the answer is correct
- n_u: number of questions not answered
- n_{aw}: the number of questions for which the answer is incorrect
- n: number of questions, thus we have $n = n_{ac} + n_{aw} + n_u$

Also a confusion matrix as Table 3.1 can be generated according to the result comparing to the ground truth of the provided dataset:

Table 3.1: Confusion Matrix

		Truth: Positive	Truth: Negative
Predicted	Positive	True Positive(TP)	False Positive(FP)
	Negative	False Negative(FN)	True Negative(TN)

ROC and AUC

Receiver operating characteristic curves (ROC curves) are usually used to illustrate the performance of a binary classification model (Fawcett, 2006). As Figure 3.1

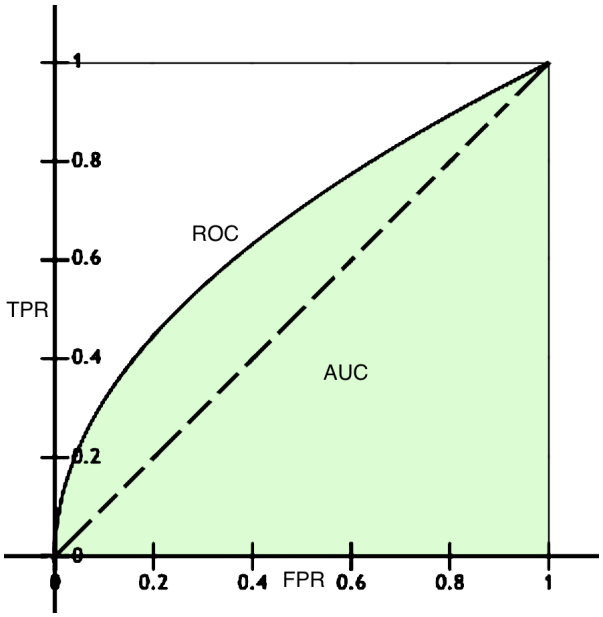

Figure 3.1: ROC space and AUC (the shade)

In Figure 3.1, we have True Positive Rate (TPR, also as recall) and False Positive Rate (FPR) defined as:

$$TPR = recall = \frac{TP}{TP + FN},\ FPR = \frac{FP}{TN + FP} \tag{3.5}$$

In a binary classification problem, the coordinate representing the performance of classification changes by selecting different standard or threshold of classification when we have the possibilities for individuals. The ROC curves of the classification and the areas under curve (AUCs) can be used to compare different classification approaches.

Non-Response Measurement c@1

There are situations where it is better not to response for systems to make judgement. An accuracy measure with non-responses is proposed by Peñas and Rodrigo (Peñas and Rodrigo, 2011), called c@1, with a balance of stability, sensitivity properties and

discrimination power. The definition of $c@1$ is provided as Eq.3.6.

$$c@1 = \frac{n_{ac}}{n} + \frac{n_{ac}}{n}\frac{n_u}{n} = \frac{1}{n}(n_{ac} + \frac{n_{ac}}{n}n_u) \tag{3.6}$$

The higher $c@1$ score is, the better the classification performs.

3.2 Steps

3.2.1 Part-of-Speech Tagging

Part-of-Speech (POS) tagging operations are implemented by RNNTagger (Schmid, 2019b,a). Tokens are separated from texts and tagged with POS types and lemmas. Empty lines are also considered and tagged as "NEWLINE", which means the n-grams generated will leap the newlines between paragraphs.

One problem I found during experiments is that the double dashes "−−" with no spaces between words have a low tagging accuracy, they should be tagged as "`:`". So I have a pre-processing step before POS tagging by adding spaces before and after "−−"

3.2.2 Generating Observations

I generate observations of texts based on sliding windows. Various types of n-grams as well as other kinds of characteristics are considered. Since the numbers of documents in each problem sets are small, I just calculate the tfs instead of tf-idfs of the terms.

Observation Types

The original list of observations are from the configuration files in the work of Moreau et al. (2014, 2015) with customization applied. The configurations for observation generating is huge in the previous work. Since we are not focusing on the optimization, the original configuration is simplified.

- Word n-gram without stopword filtering. $n \in 1, 2, 3$.
- Word n-gram with stopword filtering (with 2 stopword lists). $n = 2, 3$.
- Stopword n-gram (with 2 stopword lists). $n = 2, 3$.

- Character n-gram. $n = 4, 5$.
- Word length n-gram, with words classified by token length. $n = 2, 3$.
  - Length classes: 1-2, 3-4, 5-6, 7-8, 9-10, 11 and more.
- N-grams with mixed with tokens, lemmas, POS tags and skips. A table of this configuration can be found as Table 3.2. Notice that here I extend the mixed n-grams with skips:
  - T: token.
  - L: lemma.
  - P: POS tag.
  - S: skip, which also means any token.
- Token-type ratio (TTR), the number of distinct tokens divided by total number of words, so it represents the diversity of tokens in a text.
- Token-type term frequencies, including token with all upper case letters, all lower case letters, with only the first letter in upper case, mixed cases, numbers, punctuation, newlines and other uncategorized tokens aggregated as "misc".

Table 3.2: Mixed N-gram Configurations

Length	**Without Skips**	**With Skips**
1	P, L	
2	PP, PT, TP	
3	PPP, PPT, PTT, TPP, TTP, LSL	PSP, TST, LSL
4	PPPP	PPST, TSPP, LSSL
5		PSSSP, LSSSL, LSSSP, PSPSP, LSLSL, LSPSP
6		LSSPSP, PSSPSP, LSPSSP, PSPSSP

Pre-defined stopword lists used in this work can be found in **Appendix A** or in the source code.

The absolute count is calculated according to the most kind of observations (n-grams). All the tokens are converted to lower cases before generating n-grams.

Observation Filtering

In the works from Moreau et al. (2014, 2015), observations are filtered based on these settings:

- min absolute count: the count of one kind of observations in a text should be more or equal to: 2, 3, 5.
- min occurrence across all documents: 10%, 25%, 50% of all the documents has this kind of observations.
- min occurrence across known documents: 30%, 50% of the known documents has this kind of observations.

I did a cross validation using general classifiers before the main experiment based on the dataset from PAN 2015 (Stamatatos et al., 2015). Since it's a quick test, features and process used are quite different from the method introduced in this chapter. In a K-fold (5-fold) cross validation with 50 repeats using 4 classifiers: SVMs with linear kernel, polynomial kernel and RBF kernel, and Random Decision Tree. Top 20 classifiers are kept and their average performance ($AUC * c@1$) is calculated.

Since it is not the main part of the experiment, observations used in this test are simplified, partial result statistics can be found in Table 3.3. Where "v" is the min absolute count, "a" is the min occurrence across all documents, "k" is the min occurrence across known documents.

Table 3.3: Cross Validation on the Observation Filtering settings (avg. $AUC * c@1$)

Obs. Filter. Param.	**Without Introduced Features**	**With Introduced Features**
v=3, a = 0.1, k = 0.3	0.246	0.326
v=3, a = 0.1, k = 0.5	0.220	0.224
v=3, a = 0.25, k = 0.3	0.238	0.224
v=3, a = 0.25, k = 0.5	0.228	0.316
v=3, a = 0.5, k = 0.3	0.210	0.226
v=3, a = 0.5, k = 0.5	0.218	0.248

We can see a trend of downgrade when filtering more features out. To simplify the comparison, I just use this group of settings with several tries on different settings:

- min absolute count: 3.
- min occurrence across all documents: 10%.
- min occurrence across known documents: 30%.

To notice that termset features will be affected, when a related uni-gram is filtered.

I calculate the weighted termset features based on word uni-grams after observation filtering. Mainly following the method explain in the work of Badawi and Altınçay (2014), but the relevance frequency calculation using Eq.2.1 is modified, since in I do not have many known and unknown documents in a single problem. The original relevance frequencies are calculated based on file counts, in this work, I use term frequencies instead:

For term t_i, t_j with their tfs $tf_{i,k}, tf_{j,k}$ in the known document and $tf_{i,u}, tf_{j,u}$ in the unknown document (token count: $lenU$):

- $A = tf_{i,k} \times tf_{j,k}$
- $P = tf_{i,k} - A.$
- $R = tf_{j,k} - A.$
- $C = tf_{i,u} \times tf_{j,u}.$
- $Q = tf_{i,u} - C.$
- $S = tf_{j,u} - C.$

Then we have the relevance frequency RF:

$$RF(t_i, t_j) = \begin{cases} log_2(2 + \frac{A}{max(C,1/lenU)}) & \text{both } t_i \text{ and } t_j \text{ occur} \\ log_2(2 + \frac{P}{max(Q,1/lenU)}) & t_i \text{ occurs but not } t_j \\ log_2(2 + \frac{R}{max(S,1/lenU)}) & t_j \text{ occurs but not } t_i \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

The basic idea is to use term frequencies to represent the possible number of documents satisfying the conditions.

3.2.3 General Classifiers

The experiments of general classifiers contain two parts: generating pairwise features and training.

Generating Features

Pairwise features are generated considering several aspects for training general classifiers. I consider distance, consistency, divergence and confidence aspects of observations described in the work of Moreau et al. (2014):

- **Distance**: absolute distance of each term frequency is calculated, also Euclidean distances, cosine distances and Pearson distances are calculated base observation genres (e.g. uni-gram and bi-gram as two different genres).
- **Consistency**: To measure how stable one observation behaves in the known and unknown texts. I just use the absolute distance of one observation between known and unknown texts.
- **Divergence**: Inspired by the Jaccard similarity Eq.3.4, I use Eq.3.8 to measure the divergence of the documents, which means to what extent a particular observation is specific to an author. The document level of divergence is calculated as:

$$J_K = \frac{p+q}{p+q+r}, J_U = \frac{p+r}{p+q+r} \tag{3.8}$$

where p is the number of words found in both known (K) and unknown (U) texts, q is the number of words found in the known text but not in the unknown text and r is the number of words found in the unknown text but not in the known text. For word level divergence, I calculate the divergence for term t_i with Eq.3.9:

$$d_{i,K} = \begin{cases} J_K \times tf_i & \text{no } t_i \text{ in U} \\ 0 & t_i \text{ occurs in U} \end{cases} \quad d_{i,U} = \begin{cases} J_U \times tf_i & \text{no } t_i \text{ in K} \\ 0 & t_i \text{ occurs in K} \end{cases}, \tag{3.9}$$

- **Confidence**: Confidence is calculated to identify the most discriminative observations. Word level confidences are measure by combining consistency and divergence values. We use the mean, geometric mean and product of consistency and divergence values.

Training

After gathering all the features, a training process with 50 repeats of k-fold ($k = 5$) validation with randomized parameters are performed. In each run, 4 classifiers: random decision trees, SVMs with linear kernel, polynomial kernel and RBF kernel are trained. I collect the top 20 classifiers as well as 20 linear kernel SVMs to observe the coefficients of the linear classifiers.

In order to implement the non-response measurement, I simply use a threshold *probGap* to measure the distance of the pairwise possibilities produced from the classifiers (the possibilities of $\langle same - author \rangle$ and $\langle diff - author \rangle$). A range of values are tested to choose the best *probGap* for each of the classifiers.

3.2.4 The Impostor Method

The Impostor method I am using comes from Potha and Stamatatos (2017).

Algorithm 3: The Improved General Impostors method

input : $\langle K, U \rangle$: A pair of documents, Known and Unknown.
S: A set of impostors.
$rate\%$, k, n, m, Δ^*.

output: $\langle same - author \rangle$ or $\langle diff - author \rangle$

Set $Score = 0$;

Select m impostors as $S_{selected}$ from S with top m similarities with K;

for *i=1* **to** k **do**

Randomly select $rate\%$ of the features from the full feature pool;

Randomly select n impostors $I_1, ..., I_n$ from $S_{selected}$;

Calculate $sim(:)_{impostors}$: the similarities for n selected impostors $I_1, ..., I_n$ with U;

Calculate $rank$: Rank $sim(K, U)$ in $sim(:)_{impostors} \cup sim(K, U)$ in decreasing order;

$Score = Score + 1/(k * rank)$;

end

Return $\langle same - author \rangle$ **if** $Score > \Delta^*$; **else** $\langle diff - author \rangle$

The impostor set S I am using is a partial set of news articles (Sharif, 2018). The selected corpus contains 787 texts (average token counts ≈ 495). The topics are mainly covering entertainment and technology.

We use MinMax similarity Eq.3.3 and cosine similarity Eq.3.2 to measure the similarities between texts in this method. The features used here is simplified by using just tfs and weighted termset frequencies but no other statistics comparing the feature generation part before the general classifiers.

For each of the known texts, we select $m = 30$ impostors with top similarities. For each of the $k = 50$ repetitions, $n = 8$ impostors are selected randomly to calculate and compare the similarities with the known and unknown texts. Scores are aggregated and a best threshold is found during the training process.

3.2.5 Universum Inference

I am using the adapted Universum Inference method described in Chapter 2 (Moreau et al., 2015). For each pair of known and unknown texts, $n = 40$ repetition are calculated and the average score is recorded. As I mentioned in the previous section of the Impostor method, the similarities used here are also MinMax similarities Eq.3.3 and cosine similarities Eq.3.2, with the same set of simplified features.

From Algo.2, I intuitively think that there are relation ships between the similarities calculated, if K and U are from different authors:

$$\begin{aligned} & sim(K_1, K_2) \approx sim(U_1, U_2) \approx sim(K_3' \cap U_3', K_3'' \cap U_3'') \\ > & sim(K_1, K_3' \cap U_3') \approx sim(K_2, K_3' \cap U_3') \approx sim(K_1, K_3'' \cap U_3'') \approx sim(K_2, K_3'' \cap U_3'') \\ \approx & sim(U_1, K_3' \cap U_3') \approx sim(U_2, K_3' \cap U_3') \approx sim(U_1, K_3'' \cap U_3'') \approx sim(U_2, K_3'' \cap U_3'') \\ > & sim(K_1, U_1) \approx sim(K_1, U_2) \approx sim(K_2, U_1) \approx sim(K_2, U_2) \end{aligned} \tag{3.10}$$

So the 15 similarities are divided into 3 groups. Similarities are sorted and relative positions are compared between each similarities from different groups, the ranking of generated similarities should remain the same position, e.g. $score = score + 1$ when $sim(K_2, U_2)$ is one of the top K smallest similarities. Higher the score, lower the confusion, more possibility to be assigned as $< diff-author >$, which is the **opposite** comparing the scoring in Algo.2.

Chapter 4

Experiments

The experiments in this work are performed on 3 separate datasets, which all come from the PAN Authorship Verification shared tasks. The tasks are classification problems, in which pairs of known and unknown texts are given and the aim is to judge if the texts are from the same author. I choose these 3 problem settings since they are organized in a similar way but different in details. Full introductions to each dataset are present in each section.

4.1 PAN 2013

I am using dataset by Juola and Stamatatos (2013) in this experiment. The dataset covers English, Greek, and Spanish. For experiment purpose I only take the English part. Best efforts are made to assure that texts within a problem set are matched for genre, register, theme, and date of writing. The corpus has 18 training cases and 28 test cases. After hash value filtering, there are 16 known texts and 10 unknown texts in the training dataset (avg. token count: 1220), 67 known texts and 30 unknown texts in the test dataset (avg. token count: 1224).

4.1.1 Result

General Classifiers

Table 4.1: PAN2013 - General Classifiers Result

Setting	**Accuracy avg.**	**Prob. Gap avg.**	**c@1 avg.**	**AUC*c@1 avg.**
Universal n=20				
Original Obs. Settings	0.562	0.000	0.562	0.317
Plus Skip N-grams	0.499	0.000	0.499	0.251
Plus Termsets	0.561	0.000	0.561	0.315
With All	0.506	0.000	0.506	0.257
SVM with linear kernel n=20				
Original Obs. Settings	0.572	0.000	0.572	0.326
Plus Skip N-grams	0.539	0.000	0.539	0.289
Plus Termsets	0.537	0.000	0.537	0.287
With All	0.577	0.000	0.577	0.331

Table 4.2: PAN2013 - Linear (abs.) Coefficients Top 20% (of 28205)

Type	**Count**	**Avg.**	**c@1 Median**	**Std.**
Universal	5641	0.6020	0.5844	0.0599
Skip N-grams	1162	0.5913	0.5763	0.0491
Termsets	**2271**	**0.6195**	0.6027	0.0685

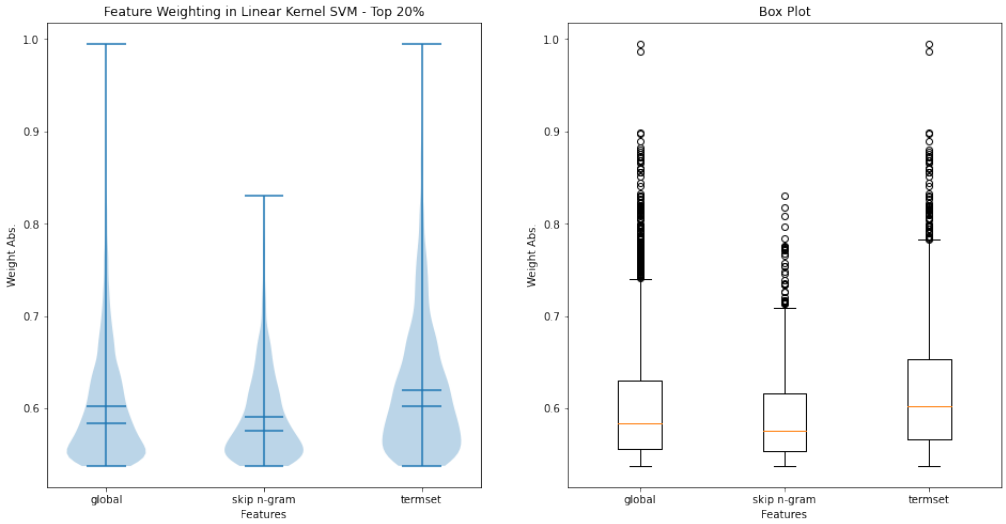

Figure 4.1: PAN2013 - Linear Kernel SVMs (abs.) Coefficients Top 20%

The Impostor Method

Table 4.3: PAN2013 - the Impostor Result

Setting	**Training Thres. & Acc.**	**Test Acc.**	**Test Best Thres. & Acc.**
Original Obs. Settings	0.6133 - 0.700	0.567	0.6378 - 0.600
Plus Skip N-grams	0.3465 - 0.700	0.433	0.3908 - 0.600
Plus Termsets	0.3016 - 0.600	0.500	0.2042 - 0.700
With All	0.2279 - 0.700	0.500	0.2629 - 0.600

Universum Inference

Table 4.4: PAN2013 - Universum Inference Result

Setting	**Training Thres. & Acc.**	**Test Acc.**	**Test Best Thres. & Acc.**
Original Obs. Settings	9.215 - 0.900	0.733	8.795 - 0.800
Plus Skip N-grams	8.890 - 0.800	0.733	8.509 - 0.767
Plus Termsets	8.890 - 0.800	0.733	8.5093 - 0.7666
With All	8.577 - 0.800	0.833	8.505 - 0.867

4.2 PAN 2014 - Essays

The English essay part of Stamatatos et al. (2014) is a set of essays from English-as-second-language learners. Two test sets are provided, and I take the second one in my experiments. There are both 210 problems for training and testing. After hashing the text files, the training corpus contains 476 known texts and 192 unknown texts (avg. token count: 973). The training corpus contains 476 known texts and 192 unknown texts (avg. token count: 973). The test problems set has 469 known texts and 190 unknown texts (avg. token count: 956).

4.2.1 Result

General Classifiers

Table 4.5: PAN2014 Essays - General Classifiers Result

Setting	**Accuracy avg.**	**Prob. Gap avg.**	**c@1 avg.**	**AUC*c@1 avg.**
Universal n=20				
Original Obs. Settings	0.520	0.073	0.423	0.226
Plus Skip N-grams	0.617	0.323	0.599	0.378
Plus Termsets	0.659	0.463	0.592	0.400
With All	0.615	0.325	0.607	0.386
SVM with linear kernel n=20				
Original Obs. Settings	0.660	0.448	0.594	0.401
Plus Skip N-grams	0.614	0.308	0.604	0.381
Plus Termsets	0.664	0.445	0.601	0.407
With All	0.613	0.288	0.612	0.386

Table 4.6: PAN2014 Essays - Linear (abs.) Coefficients Top 20% (of 22741)

Type	**Count**	**Avg.**	**c@1 Median**	**Std.**
Universal	4548	0.5787	0.5639	0.0512
Skip N-grams	577	0.5890	0.5680	0.0628
Termsets	**2773**	**0.5795**	0.5658	0.0494

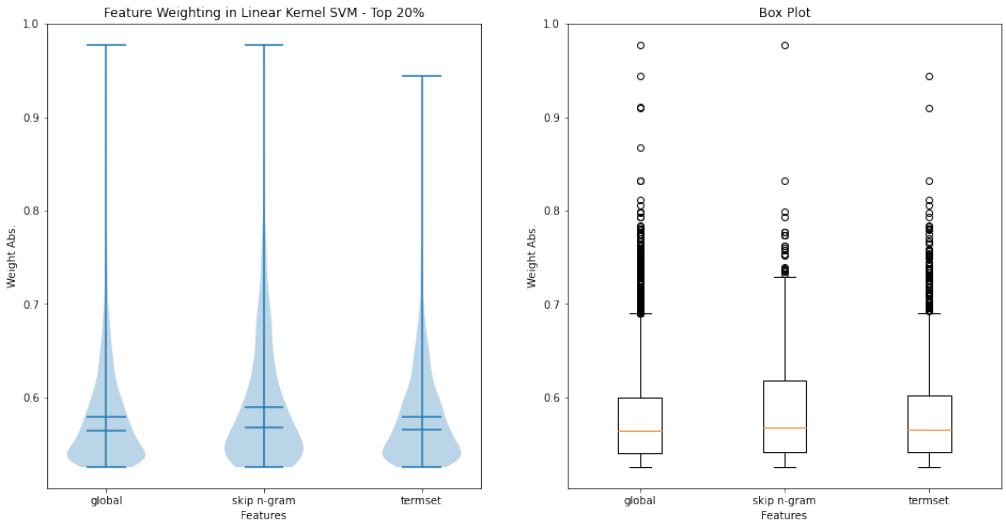

Figure 4.2: PAN2014 Essays - Linear Kernel SVMs (abs.) Coefficients Top 20%

The Impostor Method

Table 4.7: PAN2014 Essays - the Impostor Result

Setting	**Training Thres. & Acc.**	**Test Acc.**	**Test Best Thres. & Acc.**
Original Obs. Settings	0.6802 - 0.555	0.540	0.7738 - 0.620
Plus Skip N-grams	0.3271 - 0.545	0.525	0.3795 - 0.600
Plus Termsets	0.2226 - 0.555	0.485	0.2657 - 0.565
With All	0.2012 - 0.555	0.505	0.2406 - 0.605

Universum Inference

Table 4.8: PAN2014 Essays - Universum Inference Result

Setting	**Training Thres. & Acc.**	**Test Acc.**	**Test Best Thres. & Acc.**
Original Obs. Settings	10.003 - 0.565	0.550	11.090 - 0.590
Plus Skip N-grams	9.620 - 0.580	0.550	9.945 - 0.565
Plus Termsets	7.530 - 0.545	0.530	8.280 - 0.595
With All	10.448 - 0.535	0.525	8.310 - 0.635

4.3 PAN 2014 - Novels

The English novel part of Stamatatos et al. (2014) is a set of "Cthulhu Mythos" fictions. The training corpus contains 476 known texts and 192 unknown texts (avg. token count: 973). Two test sets are provided, and I take the second one (as I do for the essay part). There are 104 training problems and 204 test problems. After filtering by hashing, the training corpus contains 24 known texts and 23 unknown texts (avg. token count: 3670). The test problems consist of 23 known texts and 20 unknown texts (avg. token count: 7220).

4.3.1 Result

General Classifiers

Table 4.9: PAN2014 Novels - General Classifiers Result

Setting	**Accuracy avg.**	**Prob. Gap avg.**	**c@1 avg.**	**AUC*c@1 avg.**
Universal n=20				
Original Obs. Settings	0.651	0.000	0.651	0.423
Plus Skip N-grams	0.651	0.000	0.651	0.425
Plus Termsets	0.648	0.000	0.648	0.420
With All	0.655	0.018	0.657	0.433
SVM with linear kernel n=20				
Original Obs. Settings	0.637	0.055	0.638	0.411
Plus Skip N-grams	0.618	0.140	0.607	0.385
Plus Termsets	0.633	0.088	0.631	0.404
With All	0.626	0.178	0.620	0.400

Table 4.10: PAN2014 Novels - Linear (abs.) Coefficients Top 20% (of 107837)

Type	**Count**	**Avg.**	**c@1 Median**	**Std.**
Universal	21567	0.6180	0.6180	0.0369
Skip N-grams	3401	0.6206	0.6109	0.0376
Termsets	**11507**	**0.6150**	0.6061	0.0332

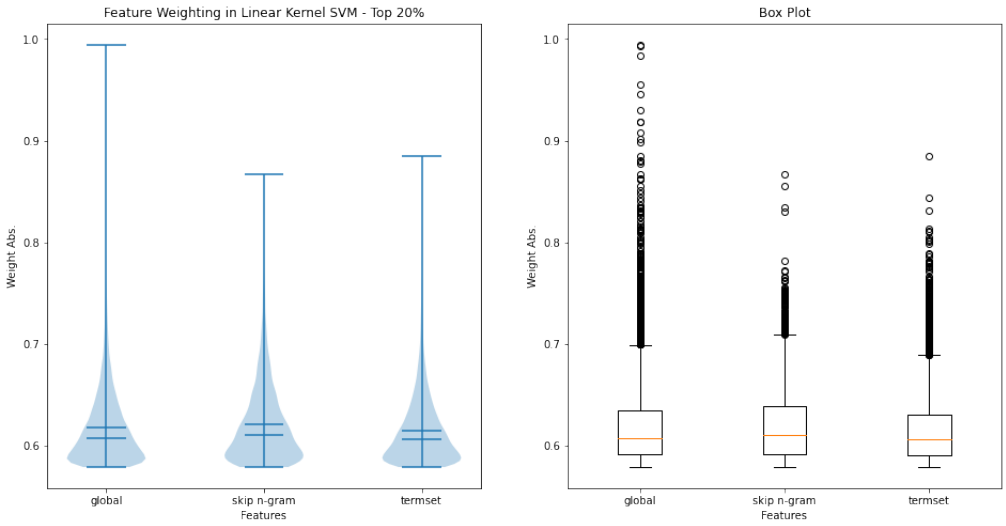

Figure 4.3: PAN2014 Novels - Linear Kernel SVMs (abs.) Coefficients Top 20%

The Impostor Method

Table 4.11: PAN2014 Novels - the Impostor Result

Setting	**Training Thres. & Acc.**	**Test Acc.**	**Test Best Thres. & Acc.**
Original Obs. Settings	0.9690 - 0.540	0.490	1.065 - 0.505
Plus Skip N-grams	0.4782 - 0.570	0.460	0.2879 - 0.505
Plus Termsets	0.2520 - 0.620	0.470	0.4829 - 0.510
With All	0.3660 - 0.560	0.470	0.2009 - 0.515

Universum Inference

Table 4.12: PAN2014 Novels - Universum Inference Result

Setting	**Training Thres. & Acc.**	**Test Acc.**	**Test Best Thres. & Acc.**
Original Obs. Settings	10.003 - 0.565	0.550	11.090 - 0.590
Plus Skip N-grams	9.970 - 0.590	0.490	5.834 - 0.565
Plus Termsets	7.530 - 0.545	0.530	8.280 - 0.595
With All	10.448 - 0.535	0.525	8.310 - 0.635

4.4 PAN 2015

I am using dataset by Stamatatos et al. (2015) in this experiment. The dataset is divided into four parts: Dutch, Greek, Spanish and English. For my aims only consider English part of the dataset. Texts from English part is cross-topic. The known texts are cross-topic play scripts of actors speaking on stage. Though there are 100 pairs of training texts and 500 pairs of test texts. After filtering by hash value of the texts, there are only 78 known texts with 37 unknown texts in training data (avg. token count: 366), and 22 known texts with 61 unknown texts (avg. token count: 536) in test data.

4.4.1 Result

General Classifiers

Table 4.13: PAN2015 - General Classifiers Result

Setting	**Accuracy avg.**	**Prob. Gap avg.**	**c@1 avg.**	**AUC*c@1 avg.**
Universal n=20				
Original Obs. Settings	0.520	0.073	0.423	0.226
Plus Skip N-grams	0.564	0.070	0.382	0.214
Plus Termsets	0.509	0.078	0.408	0.212
With All	0.567	0.105	0.389	0.226
SVM with linear kernel n=20				
Original Obs. Settings	0.516	0.105	0.375	0.199
Plus Skip N-grams	0.566	0.113	0.363	0.206
Plus Termsets	0.505	0.090	0.372	0.198
With All	0.562	0.113	0.397	0.229

Table 4.14: PAN2015 - Linear (abs.) Coefficients Top 20% (of 9346)

Type	**Count**	**Avg.**	**c@1 Median**	**Std.**
Universal	1869	0.6456	0.6315	0.0524
Skip N-grams	296	0.6553	0.6429	0.0623
Termsets	**977**	**0.6441**	0.6331	0.0481

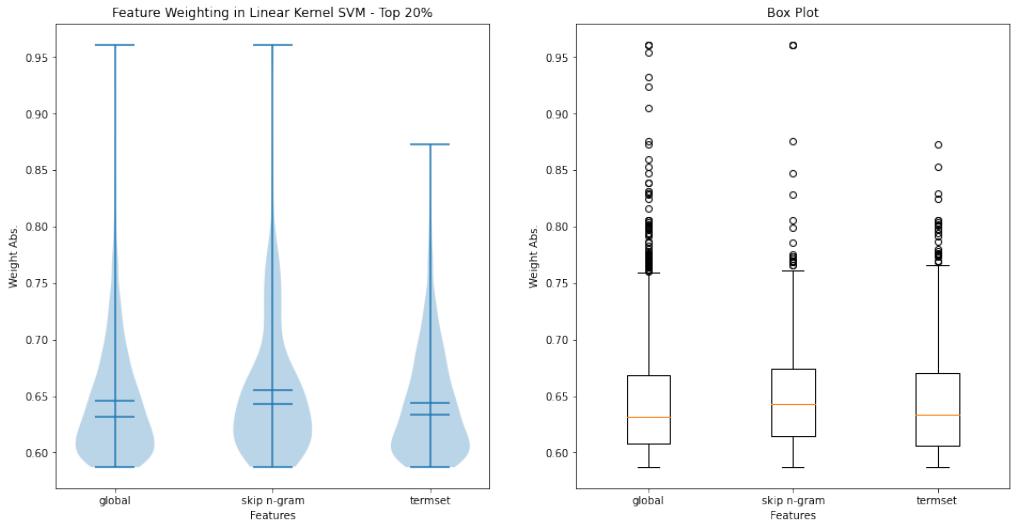

Figure 4.4: PAN2015 - Linear Kernel SVMs (abs.) Coefficients Top 20%

The Impostor Method

Table 4.15: PAN2015 - the Impostor Result

Setting	**Training Thres. & Acc.**	**Test Acc.**	**Test Best Thres. & Acc.**
Original Obs. Settings	1.047 - 0.590	0.512	1.1326 - 0.560
Plus Skip N-grams	0.5323 - 0.670	0.508	0.6055 - 0.534
Plus Termsets	0.3650 - 0.560	0.498	0.4290 - 0.542
With All	0.4332- 0.570	0.508	0.3970 - 0.512

Universum Inference

Table 4.16: PAN2015 - Universum Inference Result

Setting	**Training Thres. & Acc.**	**Test Acc.**	**Test Best Thres. & Acc.**
Original Obs. Settings	8.013 - 0.600	0.550	8.800 - 0.570
Plus Skip N-grams	7.700 - 0.640	0.548	9.570 - 0.564
Plus Termsets	7.825 - 0.670	0.572	9.940 - 0.592
With All	8.060 - 0.650	0.586	8.140 - 0.596

4.5 Overall Observation

I try to determine the *probGap* for non-response in the experiments, but the performance is low. So I did not including the non-response classification in the Impostor and Universum Inference experiments.

Universal Inference outperforms the Impostor method globally, but their performances are all lower than expected (except PAN 2013). It might be that the Impostor method and Universum Inference I implement in the experiments are not completely correct. We can see gaps between the actual thresholds of test problems and training best thresholds. Another reason for the Impostor method with low performance is that the set of impostors are not finely generated. I thought there is a filtering for impostors with top similarities for each known texts before scoring, so a prepared external source is used (Sharif, 2018). The impostors genre difference can affect the performance (Koppel and Winter, 2014). Universum Inference should be performed on a larger set of texts in a same genre, more repetitions should be made on the scoring stage.

Skip n-grams in some cases lower the performance of the classifications, which is not what I expected. The reason might be the skip n-gram, instead of concrete POS n-grams, aggregate the observations since the skips can conclude whatever tokens. So the weight in linear kernel SVMs are high, but they do not contribute much to the improvement.

But in PAN 2015, the influences caused by termset features are less than skip n-gram features. The reason might be that the PAN 2015 dataset is close to oral form of expression, since they are play scripts. More termsets features including stopwords are captured during the feature generation, and may cause the performance to downgrade.

For general classifiers. In most of the top 20 selected classifiers are SVMs with linear kernels (PAN2013: 8/20, PAN2014 essays: 16/20, PAN2014 novels: 10/20, PAN2015: 14/20). When the average token counts get larger, random decision trees occur more often since the features are getting larger.

Longer texts can get better results. The PAN 2013, PAN 2014 novel part and essay part get the good result comparing the PAN 2015 problems. The universe of PAN 2013 and PAN 2014 is narrower because the genre and topic is limited. PAN 2015 problems, since they are play scripts, are more oral than in written style. The average length is also smaller, which means it contains less features and less counts for each

of the features. Perhaps the observation filtering strategy using minimum count of observations can cause harm to the performance.

Chapter 5

Conclusion and Future Works

5.1 Conclusion

I completed the English parts of Authorship Verification task from PAN 2013, PAN 2014 (novels and essays) and PAN 2015. It shows that the results from 4 tasks differ on the dataset given and these factors have impact on the verification process.

A large number of features are generated by various types of n-grams and other observations. I adapt the existing termset weighting method to a modified version for small set of texts under Authorship Verification task settings by considering the term frequencies instead of the numbers of documents containing such terms.

Cross validations are mode to select best classifiers from trained SVMs and random decision trees classifiers. Linear kernel SVMs are recorded separately for observations of features with top 20% absolute coefficients.

Optional strategies are also implemented. An Impostor method with detailed scoring and impostor filtering is implemented, though the performance is lower than expected. The impostors set may has impact on the performance, since the genre of impostors and overall similarities between the selected impostors and known texts are not ensured in this work.

A modified version of Universum Inference method is also implemented with less chunks of files and simplified scoring. The performance among 4 datasets is unstable, perhaps having more more repetitions and more chunks of texts divided in the process will increase the performance. It may also improve the performance if the average text

length of the corpus can be larger.

The performance of skip n-gram and termset features varies among the selected tasks. The skip n-gram features in some cases lower the performance when they are added into classification process, the reason might be the vague capture of a skip n-gram aggregates several concrete POS n-grams, this kind of aggregation does not contribute to the improvement of the performance. The termset features have majorities of the features with top absolute coefficient rankings, while the skip n-gram features tend to perform better under oral environment than the termset features.

5.2 Future Works

The source code is implemented but not optimized and utilized. Configurations need to be isolated from the implementation and optimization on the calculation is also required. The time consumed by a complete run on the 4 dataset is huge (e.g. approx. 32 hours for the Universum Inference method, with multiple processes on a machine with 8 cores).

The Impostor method and Universum Inference did not achieve the expected performance level. The adaption under Authorship Verification settings may cause the downgrade of the algorithms. The original versions of two strategies need to be implemented under Authorship Verification tasks.

From the years of overviews from authorship related tasks hosted by PAN, neural networks, especially RNNs, are taking lead of the methods used in the classification problem. Because of the limited schedule and ability, I did not make comparison between with and without skip n-grams and termset observations in a neural network based classification process.

A new round of shared task on Authorship Verification is just held by PAN in 2020 (Bevendorff et al., 2020). The corpus used in PAN 2020 is a set of fan fictions, with their meta data of classification on the source website (FanFiction.Net, 2019). Metrics such as AUC, F1-score, c@1 and F_0.5u (Bevendorff et al., 2019) are introduced in the new shared task. A group of improved experiments should be implemented based on PAN 2020 Authorship Verification problem set.

Bibliography

Asif, M. S. and J. Romberg (2013). Sparse Recovery of Streaming Signals Using L1-Homotopy.

Axelsson, M. W. (2000). USE-the Uppsala Student English corpus: an instrument for needs analysis. *ICAME journal 24*, 155–157.

Badawi, D. and H. Altınçay (2014). A novel framework for termset selection and weighting in binary text classification. *Engineering Applications of Artificial Intelligence 35*, 38–53.

Bagnall, D. (2015). Author identification using multi-headed recurrent neural networks. *arXiv preprint arXiv:1506.04891*.

Bevendorff, J., B. Ghanem, A. Giachanou, M. Kestemont, E. Manjavacas, M. Potthast, F. Rangel, P. Rosso, G. Specht, E. Stamatatos, B. Stein, M. Wiegmann, and E. Zangerle (2020). Shared tasks on authorship analysis at pan 2020. In J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, and F. Martins (Eds.), *Advances in Information Retrieval*, Cham, pp. 508–516. Springer International Publishing.

Bevendorff, J., B. Stein, M. Hagen, and M. Potthast (2019, June). Generalizing Unmasking for Short Texts. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, pp. 654–659. Association for Computational Linguistics.

Bezdek, J. C., R. Ehrlich, and W. Full (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences 10*(2-3), 191–203.

Castro, D., Y. Adame, M. Pelaez, and R. Muñoz (2015). Authorship verification, combining linguistic features and different similarity functions. *CLEF (Working Notes)*.

Digman, J. M. (1990). Personality structure: Emergence of the five-factor model. *Annual review of psychology 41*(1), 417–440.

El, S. E. M. and I. Kassou (2014). Authorship analysis studies: A survey. *International Journal of Computer Applications 86*(12).

FanFiction.Net (2019). FanFiction. `https://www.fanfiction.net/`. Last accessed at 2020-09-05.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters 27*(8), 861–874. ROC Analysis in Pattern Recognition.

Figueiredo, F., L. Rocha, T. Couto, T. Salles, M. A. Gonçalves, and W. M. Jr. (2011). Word co-occurrence features for text classification. *Information Systems 36*(5), 843–858.

Frery, J., C. Largeron, and M. Juganaru-Mathieu (2014, 09). UJM at CLEF in author verification based on optimized classification trees. In *Proc. Int. Conf. CLEF Notebook PAN*, pp. 1042–1048.

Guthrie, D., B. Allison, W. Liu, L. Guthrie, and Y. Wilks (2006, May). A Closer Look at Skip-gram Modelling. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Gutierrez, J., J. Casillas, P. Ledesma, G. Fuentes, and I. Meza (2015). Homotopy based classification for author verification task. *Working Notes Papers of the CLEF*.

HaCohen-kerner, Y., Z. Ido, and R. Ya'akobov (2017). Stance Classification of Tweets Using Skip Char Ngrams. In Y. Altun, K. Das, T. Mielikäinen, D. Malerba, J. Stefanowski, J. Read, M. Žitnik, M. Ceci, and S. Džeroski (Eds.), *Machine Learning and Knowledge Discovery in Databases*, Cham, pp. 266–278. Springer International Publishing.

Iqbal, F., H. Binsalleeh, B. C. Fung, and M. Debbabi (2010). Mining writeprints from anonymous e-mails for forensic investigation. *Digital Investigation 7*(1), 56–64.

Juola, P. (2006, December). Authorship Attribution. *Found. Trends Inf. Retr. 1*(3), 233–334.

Juola, P. and E. Stamatatos (2013, September). PAN13 Author Identification: Verification [Dataset].

Keogh, E., S. Lonardi, and C. A. Ratanamahatana (2004). Towards Parameter-Free Data Mining. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, New York, NY, USA, pp. 206–215. Association for Computing Machinery.

Kešelj, V., F. Peng, N. Cercone, and C. Thomas (2003). N-gram-based author profiles for authorship attribution. In *Proceedings of the conference pacific association for computational linguistics, PACLING*, Volume 3, pp. 255–264. sn.

Khonji, M. and Y. Iraqi (2014). A Slightly-modified GI-based Author-verifier with Lots of Features (ASGALF). *CLEF (Working Notes) 1180*, 977–983.

Koppel, M., J. Schler, S. Argamon, and Y. Winter (2012). The "fundamental problem" of authorship attribution. *English Studies 93*(3), 284–291.

Koppel, M. and Y. Winter (2014). Determining if two documents are written by the same author. *Journal of the Association for Information Science and Technology 65*(1), 178–187.

Layton, R., P. Watters, and R. Dazeley (2013, 01). Local n-grams for author identification: Notebook for PAN at CLEF 2013. *CEUR Workshop Proceedings 1179*.

Malmasi, S., M. Zampieri, and M. Dras (2016, June). Predicting Post Severity in Mental Health Forums. In *Proceedings of the Third Workshop on Computational Linguistics and Clinical Psychology*, San Diego, CA, USA, pp. 133–137. Association for Computational Linguistics.

Meretakis, D. and B. Wüthrich (1999). Extending naïve bayes classifiers using long itemsets. In *Proceedings of the Fifth ACM SIGKDD International Conference on*

Knowledge Discovery and Data Mining, KDD '99, New York, NY, USA, pp. 165–174. Association for Computing Machinery.

Modaresi, P. and P. Gross (2014). A Language Independent Author Verifier Using Fuzzy C-Means Clustering. In *CLEF (Working Notes)*, pp. 1084–1091.

Moreau, E., A. Jayapal, G. Lynch, and C. Vogel (2015, September). Author Verification: Basic Stacked Generalization Applied To Predictions from a Set of Heterogeneous Learners - Notebook for PAN at CLEF 2015. In L. Cappellato, N. Ferro, G. J. F. Jones, and E. SanJuan (Eds.), *CLEF 2015 - Conference and Labs of the Evaluation forum*, CEUR Workshop Proceedings, Toulouse, France. CEUR.

Moreau, E., A. Jayapal, and C. Vogel (2014, September). Author Verification: Exploring a Large set of Parameters using a Genetic Algorithm - Notebook for PAN at CLEF 2014. In L. Cappellato, N. Ferro, M. Halvey, and W. Kraaij (Eds.), *Working Notes for CLEF 2014 Conference*, Volume 1180, Sheffiled, United Kingdom, pp. 12. CEUR Workshop Proceedings.

Peñas, A. and A. Rodrigo (2011). A simple measure to assess non-response. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, USA, pp. 1415–1424. Association for Computational Linguistics.

Pokou, Y. J. M., P. Fournier-Viger, and C. Moghrabi (2016). Authorship attribution using small sets of frequent part-of-speech skip-grams. In *International Florida Artificial Intelligence Research Society Conference*, pp. 86–91. AAAI.

Potha, N. and E. Stamatatos (2017). An Improved Impostors Method for Authorship Verification. In G. J. Jones, S. Lawless, J. Gonzalo, L. Kelly, L. Goeuriot, T. Mandl, L. Cappellato, and N. Ferro (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, Cham, pp. 138–144. Springer International Publishing.

Satyam, A., A. K. Dawn, and S. K. Saha (2014, 09). *A statistical analysis approach to author identification using latent semantic analysis*, pp. 1143–1147.

Schmid, H. (2019a). Deep Learning-Based Morphological Taggers and Lemmatizers for Annotating Historical Texts. In *Proceedings of the 3rd International Conference*

on Digital Access to Textual Cultural Heritage, DATeCH2019, New York, NY, USA, pp. 133–137. Association for Computing Machinery.

Schmid, H. (2019b). RNNTagger - a Neural Part-of-Speech Tagger. `https://www.cis.uni-muenchen.de/~schmid/tools/RNNTagger/`. Last accessed at: 2020-08-26.

Seidman, S. (2013). Authorship verification using the impostors method. In *CLEF 2013 Evaluation labs and workshop–Working notes papers*, pp. 23–26.

Sharif, P. (2018). BBC News Summary. `https://www.kaggle.com/pariza/bbc-news-summary`. Last accessed at 2020-09-01.

Stamatatos, E. (2009). A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology 60*(3), 538–556.

Stamatatos, E., D. Daelemans, W. Ben Verhoeven, P. Juola, A. López-López, M. Potthast, and B. Stein (2015, September). PAN15 Author Identification: Verification [Dataset].

Stamatatos, E., W. Daelemans, B. Verhoeven, M. Potthast, B. Stein, P. Juola, M. A. Sanchez-Perez, and A. Barrón-Cedeño (2014, September). PAN14 Author Identification: Verification [Dataset].

Stamatatos, E., W. Daelemans, B. Verhoeven, M. Potthast, B. Stein, P. Juola, M. A. Sanchez-Perez, and A. Barrón-Cedeño (2014). Overview of the author identification task at PAN 2014. In *CLEF 2014 Evaluation Labs and Workshop Working Notes Papers, Sheffield, UK, 2014*, pp. 1–21.

Stamatatos, E., W. Daelemans, B. Verhoeven, M. Potthast, B. Stein, P. Juola, M. A. Sanchez-perez, and A. Barrón-cedeño (2013). Overview of the author identification task at pan-2013.

Stamatatos, E., M. Potthast, F. Rangel, P. Rosso, and B. Stein (2015). Overview of the PAN/CLEF 2015 Evaluation Lab. In J. Mothe, J. Savoy, J. Kamps, K. Pinel-Sauvagnat, G. Jones, E. San Juan, L. Capellato, and N. Ferro (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, Cham, pp. 518–538. Springer International Publishing.

Tesar, R., V. Strnad, K. Jezek, and M. Poesio (2006). Extending the Single Words-Based Document Model: A Comparison of Bigrams and 2-Itemsets. In *Proceedings of the 2006 ACM Symposium on Document Engineering*, DocEng '06, New York, NY, USA, pp. 138–146. Association for Computing Machinery.

Veenman, C. and Z. Li (2013, 01). Authorship verification with compression features. *CEUR Workshop Proceedings 1179*.

Veenman, C. J. and D. M. J. Tax (2005). LESS: a model-based classifier for sparse subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*(9), 1496–1500.

Vogel, C., G. Lynch, and J. Janssen (2009). Universum Inference and Corpus Homogeneity. In M. Bramer, M. Petridis, and F. Coenen (Eds.), *Research and Development in Intelligent Systems XXV*, London, pp. 367–372. Springer London.

Webis group (2020). PAN at CLEF. `https://pan.webis.de/`. Last accessed at 2020-08-31.

Zamani, H., H. N. Esfahani, P. Babaie, S. Abnar, M. Dehghani, and A. Shakery (2014). Authorship Identification Using Dynamic Selection of Features from Probabilistic Feature Set. In E. Kanoulas, M. Lupu, P. Clough, M. Sanderson, M. Hall, A. Hanbury, and E. Toms (Eds.), *Information Access Evaluation. Multilinguality, Multimodality, and Interaction*, Cham, pp. 128–140. Springer International Publishing.

Zhang Yang, Zhang Lijun, Yan Jianfeng, and Li Zhanhuai (2003). Using association features to enhance the performance of naive bayes text classifier. In *Proceedings Fifth International Conference on Computational Intelligence and Multimedia Applications. ICCIMA 2003*, pp. 336–341.

Appendix A

Stopwords

A.1 Stopword List - 50

,	that	don
’	me	but
.	all	m
the	is	do
i	we	if
you	for	no
and	be	at
!	he	have
to	my	are
a	-	was
?	they	as
s	ll	like
it	on	she
--	your	can
of	what	;
t	not	see
in	with	

A.2 Stopword List - 200

,	what	us
’	not	now
.	with	know
the	don	one
i	but	little
you	m	out
and	do	good
!	if	well
to	no	yes
a	at	go
?	have	who
s	are	man
it	was	when
--	as	them
of	like	think
t	she	got
in	can	come
that	;	too
me	see	take
all	him	sir
is	d	ye
we	oh	never
for	so	love
be	this	our
he	here	right
my	her	ay
-	his	get
they	there	say
ll	up	tell
on	ve	about
your	re	then

how
ain
been
will
old
or
where
mr
day
am
would
just
sure
very
time
only
ever
christmas
from
want
back
an
down
ah
did
two
has
"
night
by
let
should
make
shall
home
men
give
look
wish
some
ice
were
em
o
had
god
fine
going
why
sea
any
poor
more
anna
----
always
after
their
must
again
great
big
bad
every
way
find
tis
could
doctor
ship
didn
away
quite
may
much
long
nice
daughter
dat
off
claus
said
lord
life
things
talk
damn
care
world
wishing
before
such
other
money
girl
children
heart
woman

best
put
dog
anything
pretty
lady
thinking
mean
high

Appendix B

Datasets

Datasets for from PAN shared tasked can be found from this url:
`https://pan.webis.de/data.html`
The doi links of PAN 2013 - 2015 datasets are:

- 2013: `https://doi.org/10.5281/zenodo.3715999`
- 2014: `https://doi.org/10.5281/zenodo.3716033`
- 2015: `https://doi.org/10.5281/zenodo.3737563`

User should contact the author first through the Zenodo (`https://zenodo.org/`) system to gain access to the datasets.

Appendix C

Source Code and Results

Source code are hosted on GitHub(`https://github.com/`) privately, in which there is a link to the data generated from the steps of the experiments. Since the data genreated is large, I failed to submit it on the system of the school. Please email me (`chenc1@tcd.ie`) for the access.

The repository link is:
`https://github.com/tannineo/CS-MSc-dissertation`

Requirements (tested only during experiments):

- Python version 3.8
- Perl version 5.32
- RNNTagger version 1.2

Other project dependencies are managed by `pipenv` version: 2020.6.2.
`https://github.com/pypa/pipenv`