GENRE INDEPENDENT AUTHORSHIP ATTRIBUTION FOR TURKISH DOCUMENTS

PhD Thesis

Hayri Volkan AGUN

Eskişehir 2019

GENRE INDEPENDENT AUTHORSHIP ATTRIBUTION FOR TURKISH DOCUMENTS

Hayri Volkan AGUN

PhD. THESIS

Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Özgür YILMAZEL

Eskişehir
Eskişehir Technical University
Graduate School of Sciences
January 2019

FINAL APPROVAL FOR THESIS

This thesis titled "Genre Independent Authorship Attribution for Turkish Documents" has been prepared and submitted by Hayri Volkan AGUN in partial fullfillment of the requirements in "Eskisehir Technical University Directive on Graduate Education and Examination" for the Degree of Doctor of Philosophy (PhD) in Computer Engineering Department has been examined and approved on 11/01/2019.

Committee Members		Signature
Member (Supervisor)	: Assoc. Prof. Dr. Özgür YILMAZEL	…….
Member	: Assist. Prof. Dr. Alper BİLGE	…….
Member	: Assist. Prof. Dr. Alper Kürşat UYSAL	.....
Member	: Assist. Prof. Dr. Muammer AKÇAY	.....
Member	: Prof. Dr. Rıfat EDİZKAN	…….

Prof. Dr. Ersin YÜCEL
Director of Graduate School of Sciences

ABSTRACT

GENRE IDEPENDENT
AUTHORSHIP ATTRIBUTION FOR TURKISH DOCUMENTS

Hayri Volkan AGUN

Department of Computer Engineering
Eskisehir Technical University, Graduate School of Sciences, January 2019

Supervisor: Assoc. Prof. Dr. Özgür Yılmazel

In this thesis, we propose a scaling algorithm using multivariate analysis for authorship attribution in different document types with heterogeneous properties. The scaling algorithm is inspired by the idea of removing the non-variable background used in capturing moving objects in image recognition systems. This algorithm consists of two steps, which are determining the source-based common features of the documents in different topics and genres and removing these common features from the document vector for uncovering the style of the authors. Authorship attribution differs from other text classification types in terms of text processing techniques. The topic, genre, and target audience affect the author's word choice, causing the author's style to blur. In this context, the author's different types of documents are scaled according to the type which the document belongs to, and the similarity between the documents by the same author or different authors is exposed.

In the thesis, classification based accuracy measurements were made by using term and character sequences on different types of documents, such as e-mails, blogs, micro messages, newspaper articles, and novel excerpts. The proposed scaling algorithm achieves the highest accuracy regardless of topic, feature set and genre in any dataset in classification based authorship attribution. In addition, scaling on only the term or character sequence features in the cross-domain and cross-genre datasets is highly competitive with the complex text processing techniques obtained by linguistic analysis.

Keywords: Authorship Attribution, Text Classification, Multivariate Scaling.

ÖZET

TÜRKÇE METİNLERDE
FARKLI JANRLARDA YAZAR BELİRLEME

Hayri Volkan AGUN
Bilgisayar Mühendisliği Anabilim Dalı
Eskişehir Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Ocak 2019

Danışman: Doç. Dr. Özgür Yılmazel

Bu tezde heterojen özelliklere sahip farklı doküman türlerinde yazar tanıma için çok değişkenli analizin kullanıldığı bir ölçekleme algoritması önerilmektedir. Bu ölçekleme algoritması görüntü tanıma sistemlerinde hareketli obje yakalamada kullanılan değişken olmayan arka planın çıkarılması fikrinden esinlenmektedir. Bu algoritma iki adımdan oluşmaktadır. Bunlar; ortak vektör yaklaşımı kullanılarak farklı konu ve janrdaki dokümanların kaynak bazlı ortak özelliklerinin saptanması ve bu ortak özelliklerin doküman vektöründen çıkartılması ile yazar stilinin belirginleştirilmesi adımlarıdır. Yazar tanıma kullanılan metin işleme teknikleri bakımından diğer metin sınıflandırma türlerinden farklıdır. Konu, janr ve hedef okuyucu kitlesi yazarın kelime seçimine etki ederek yazarın stilinin bulanıklaşmasına neden olmaktadır. Bu bağlamda yazarın farklı türdeki dokümanlarının ait olduğu türe göre ölçeklendirmesi yapılarak dokümanların aynı yazar veya farklı yazarlar arasındaki benzerliği belirginleştirilmiştir.

Tezde e-posta, internet günlükleri, mikro mesajlar, gazete yazıları, roman alıntıları gibi farklı doküman türleri üzerinde terim ve karakter dizileri kullanılarak sınıflandırma tabanlı doğruluk ölçümleri yapılmıştır. Önerilen ölçeklendirme algoritması sınıflandırma tabanlı yazar tanımada her türlü veri kümesinde konu, özellik ve janrdan bağımsız olarak en yüksek doğruluğu elde etmiştir. Ayrıca çapraz janr ve alanlar üzerine oluşturulmuş doküman kümelerinde sadece terim veya karakter dizileri üzerinde yapılan ölçekleme dilbilimsel analiz kullanılarak elde edilen karmaşık metin işleme teknikleri ile rekabet edebilir düzeydedir.

Anahtar Sözcükler: Yazar Tanıma, Metin Sınıflandırma, Çok Değişkenli Ölçekleme.

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor Assoc. Prof. Dr. Özgür Yılmazel for the continuous support of my Ph.D study and related research, for his patience, motivation, and experience.

My sincere thanks also goes to Prof. Dr.Yılmaz Kılıçaslan and Assoc. Prof. Dr. Erdinç Uzun for motivating me in this field in general with their remarks and suggestions from the very beginning.

Last but not the least, I especially would like to thank to my father for being very supportive in my doing a PhD. and to my family: my spouse, my mother and to my brothers for supporting me spiritually throughout writing this thesis and my life in general.

Hayri Volkan AGUN

11/01/2019

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "scientific plagiarism detection program" used by Eskisehir Technical University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Hayri Volkan Agun

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

ABBREVIATIONS

AA	:	Authorship Attribution
ART	:	Article Documents
BLOG	:	Blog Documents
BOW	:	Bag of Words
CNG	:	Common N-Grams
CVA	:	Common Vector Approach
DF	:	Document Frequency
DL	:	Deep Learning
DT	:	Decision Tree
DL	:	Deep Learning
IDF	:	Inverted Document Frequency
L-BFGS	:	Limited Memory Broyden–Fletcher–Goldfarb–Shanno
LOG	:	Logistic Regression
LDA	:	Linear Discriminant Analysis
MLP	:	Multilayer Perceptron
MNB	:	Multinomial Naïve Bayes
PCA	:	Principal Component Analysis
POS	:	Part of Speech
RF	:	Random Forest
TF	:	Term Frequency
TF-IDF	:	Term Frequency Inverted Document Frequency
TWE	:	Tweet Documents

1. INTRODUCTION

On the web, people use fake accounts in order to hide themselves from authority, as in the 2016 United Stated presidential elections, where a group of people used fake identities for spreading false information to change the voter opinions. Spreading false information about sensitive events in order to manipulate and control crowds against their will is known as trolling. In trolling, the only trace left is the text documents where true identity is unknown and impossible to determine with network analysis. With an increase in trolling, social media abuse which includes spreading false, provocative or confidential government messages has drawn the attention to the task of finding the true identity behind a text document.

Finding true identity of a text document is the subject of authorship attribution (AA). It is the task of identifying the author of a disputed/unknown document. Firstly, it was suggested for finding the authors of disputed books, poems and plays by Mendenhall in 19th Century [1]. Early methods included statistical analysis techniques applied on the primary elements of text, such as words, word lengths and etc. With the advancement in classification systems, these primary elements have started to be used in feature extraction for classification of documents into author categories. Nowadays, the classification approaches are divided into two tasks, referred as identification and verification [2]. In author identification, the author of the documents must be one of the known authors, and in author verification, the author of the documents is either a known author or someone else. The sets of features are common for these tasks. They are known as stylometric features. Common stylometric features are obtained by processing the function words, vocabulary richness and readability measures, punctuation marks and character n-grams [3].

The main consensus on the characteristics of stylometric features is that they have less variability in different genres and topics. [4, 5]. The number of feature sets with such characteristics is very limited, and the proposed stylometric features are usually dependent on the domain of the documents. For the web documents, increased heterogeneity of document size, genre and topic makes the problem even more difficult. To overcome these problems, the researchers have suggested increasing the orthogonality of the features either by combining new feature sets or by feature selection and dimensionality reduction methods [6–10]. Among the new feature sets, the linguistic features including part of speech tags (POS), syntax trees, and head categories of phrases

are the most notable. [11–19]. The robustness of the proposed features in different domains are tested in cross-topic, multi-topic, single-topic, cross-genre, multi-genre and single-genre datasets [20–24]. The feature selection and dimensionality reduction techniques are widely applied on these datasets [6–10, 25–27]. Although these methodologies applied on AA tasks are useful, they are mostly restricted to the general text classification methodology and are not specific contributions to the heterogeneous nature of the AA tasks.

The main objective of this thesis is to propose a methodology for reducing the negative effects caused by the heterogeneous domains in AA. Domains, such as twitter, blogs, and articles have different characteristics. For example, tweets contain emoticons, or slang language words more often than blogs and articles, and they are shorter in length. Such varying characteristics of domains are addressed in AA literature in terms of genre and topics. Genre or topic independent feature sets [28–33], increased diversity of the features, domain adaptation [32, 34] and instance sampling methods [26, 27, 35–38] have been proposed for the robustness of AA in heterogeneous domains. However, they are not widely known and are not easily adaptable to the existing classification approaches. The following assumption and hypothesis give a broader perspective on the genre, domain and source related issues in AA.

Assumption: "One of the main objectives of an author is the understanding and adoption of his/her writings by the target audience. Topic, genre, and the domain of the target readers are multi-dimensional factors that affect the term selection process of the author in such a way that the link between the style of the author and the document becomes hidden under these factors."

Hypothesis: "Removing the correlation between the text features and the domain, genre and the topic of documents uncovers the style of the author."

In this respect, the general intuition behind this hypothesis is very similar to background subtraction task suggested for object classification in image recognition systems. Background subtraction removes the steady background in order to filter out the object patterns without reducing the dimensionality. Such approaches have been applied to text classification by scaling.

In traditional scaling methods, each feature's scaling factor is independent of each other and is identical. However, when the domain of text documents is considered, the terms are conditionally dependent on each other. For these reasons, in this thesis, a

multivariate analysis technique known as common vector approach (CVA) has been proposed in computing domain representative patterns, and through removing these patterns, the document is differentiated from its domain so that author style becomes similar in and between different domains. This approach is called CV scaling. It is applicable to any classification based AA with no dependence on feature set and/or term weighting and selection method. To show the strengths of CV scaling, we experimented with bag of words (BOW) representations in two languages with different genres and topics. As a result of experimental evaluations, it has been shown that the contribution of this approach to author classification performance is mostly significant.

The CV scaling consists of domain vector computation by CVA and subtraction of common vectors of the domain from the document vector. CVA is studied in isolated speech recognition and face recognition [39–44]. In these applications, common vectors are used for representing class invariant properties. However, in AA, we use all the domain elements, such as source of the documents, genres and topics for constructing common vectors. In subtraction process, the bucketization is applied so that the similar values that fall in the same buckets are cancelled. Along with its simple use, CVs are relatively independent of the feature space and the classification method. Unlike dimensionality reduction methods, the subtraction of common properties increases diversity among documents written by different authors without reducing the information in document vector. The subtraction process is straightforward and applicable to any representation, such as document embeddings or BOW.

2. A BRIEF REVIEW

In this section, existing AA approaches are briefly summarized in terms of historical development, methodologies and datasets. A general overview of studies, applied methodologies and recent issues has been demonstrated in each subsection.

2.1. A historical perspective

AA has been investigated for a long time. Early AA approaches are referred as traditional AA, where writings of literary texts, opera plays and documents of important people are analyzed in terms of genre, gender and authorship [45, 46]. In late 1800s, Mendenhall proposed "word spectrum" for analysing authorship by word length and length histograms [1]. By Zip's law, the author analysis methods start to use the words appearing once, twice or more [47]. During early 1900s, statistical analysis techniques, such as Bayesian analysis, joint probability inference and hypothesis testing were commonly preferred [48–51]. Features, such as function words, vocabulary richness, and readability measures were proposed during this period. Their effectiveness was measured through statistical analysis techniques that measured variability and consistency [52]. Some of these studies also partitioned the documents into meaningful segments in order to acquire more evidence for statistical analysis [53].

In the late 1900s, applicability of computational methods made principal component analysis (PCA) – a multivariate analysis technique – a common method for AA in a larger scale. Early PCA methods were used for solving disputed authorship of Federalist papers and Shakespeare letters [37, 54, 55]. Thanks to PCA, first implementations of classification and clustering methods were successfully applied to AA in newspaper documents [56–58]. Advanced computing power enabled the classification methodologies to become a new standard for AA in documents with a large number of features [59, 60]. Combination of diverse feature sets, such as function words, n-grams, collocations, punctuations, and other stylometric features have been applied in classification methodology [61–63]. Most of the focus was diverted to finding important features for different AA tasks [11, 23, 64–67]. Factors, such as topicality, text length, number of documents and genre were investigated in different datasets [33, 66, 68–73]. The general assumptions on these studies suggest that style is unique and it is separated from content [74–76]. For this reason, frequent stylometric features are usually chosen from content-independent features, such as function words, punctuations, and character

n-grams. The importance of function words, punctuations and character n-grams have been shown with recent studies in different languages [68, 77–80].

2.2. Verification and Identification

Current AA tasks are divided into two problems: closed set and open set. Early studies conducted on disputed authorship were done on closed sets, where a disputed document was written by one of the known authors [58, 81, 82]. This task is referred as identification. For example, "a questioned document is either written by Shakespeare or Fletcher" is an example of identification. On the other hand, open set problems involve known and unknown authors, which means a document is written by a known author or not. Since open set problems produce a binary outcome for an author and document pair, they are referred as verification. With the interest of forensic community, current AA approaches focus mostly on verification [3].

Although the outcome and classification objectives are different for verification and identification, they use the same type of stylometric features. Verification tasks are harder than identification, and general approaches use imposter documents for increasing the negative document samples [83–85]. Verification tasks are appropriate for detecting known set of authors (terrorists, criminals and etc.) in large scale document collections, such as web documents [86, 87]. On the other hand, studies on the identification task still provide the baselines and test grounds for testing feature selection methods within different factors, such as topicality, genre and domain.

2.3. Feature Diversity

The features used in AA are generated from the vocabulary by processing training documents. Their effectiveness for AA is determined by not only their importance but also their divergence from each other. Feature diversity is an important factor for the performance of AA. In general, the independent set of features achieve robust performance in text classification. Therefore, modern AA studies have focused on combining new feature sets with well-known stylometric features. Among those studies, typed character n-grams showed significant performance gains in web documents [34, 76, 79, 80, 88]. For the web documents, the significance of adding linguistic features, such as POS tags, syntax tree depths [15, 16], syntactic dependencies [63], morphological derivations [63, 89], entity relations [90–92], discourse linking words [93],

synonyms/hyponyms [92, 94], and modality features [95, 96] is not broadly shown. Even though addition of linguistic features is shown to be helpful [12, 97], there is not any common consensus on their effectiveness [98].

Combining different sets of features arises new questions, some of which are normalization, scaling, term weighting and term selection. In general, the common approach for combining different sets of features is to pre-process them separately and concatenating the results [17, 18, 99]. Other approaches, such as enriching a feature with meta tags [12, 28, 100] or combining them by heuristic approaches [15, 94, 101, 102] are also preferred by the AA community.

2.4. Feature Selection

Recent studies have measured the effects of term selection methods for AA in different datasets. Instead of manually adding or removing features, term selection approaches automatically score features in terms of importance and discriminative power. Later, this scoring is used for filtering the most relevant features. Among many term selection methods, only common methods, such as chi-square, odds ratio, information gain, Gini ratio and document frequency have been experimented in AA [103–105]. In these studies, basic document frequency selection demonstrates higher performance than the best performing methods in topic categorization tasks [104, 106]. In AA tasks, the frequent features are shown to be discriminative, which is why a frequency based term selection approach known as common n-grams has been widely applied for AA [31, 107–109]. Koppel et al. suggested a feature selection approach based on the degree of replacement of frequently used terms with their synonyms [110]. He reported significant performance improvement with this scoring technique. Kestemont et al. applied cross-validation for selecting the robust features in varying genres and topics [69].

2.5. Multivariate Analysis

Multivariate analysis techniques, such as principal component analysis (PCA) and linear discriminant analysis (LDA) have been widely applied in author identification [55, 58, 67]. Early author identification approaches employed PCA for the visualization of author characteristics via plotting variables and for the author assignment through nearest neighbor method. Most of these studies have significant contributions to author identification by comparing the authors of plays and historical papers [55, 111]. Ledger

et al. used PCA for analysing important terms and compared the results with cluster analysis on linear discriminant features [55]. Holmes et al. proposed PCA for dimensionality reduction of the high frequency words for the authors of Federalist papers [111]. He compared the results with a probabilistic approach where the decision parameters are adapted by a genetic search algorithm. Binongo and Smith proposed a feature selection strategy for filtering discriminative words by looking orthogonality obtained by PCA on frequent words [37]. Early applications of PCA were successful; however, there are controversial claims about the applicability and effectiveness of these methods for author identification in a larger scale [112].

Recent approaches involving PCA and LDA have been proposed for dimensionality reduction for AA in diverse set of documents [27, 58, 113, 114]. Binongo applied PCA to the text blocks of a novel for author verification [58]. Zhang et al. applied PCA prior to LDA for author identification on news document collections. He employed nearest neighbor classification on the features extracted by LDA [114]. Kestemont et al. used PCA for dimensionality reduction and stylometric analysis in historical documents [115]. They applied PCA as an input to Burrow's delta method.

2.6. Deep Learning

Deep learning (DL) is an alternative to traditional BOW representation in applications of sentiment analysis [116], POS tagging [117], and topic categorization [118]. Unlike traditional frequency-based classification approaches, DL models capture long-range dependency relations among features, and they do not need expansive pre-processing. Until recently, several studies conducted on different genres and topics have showed DL effectiveness in AA tasks [8, 12, 86, 90, 119–122]. Hitschler et al. improved the performance of author identification with multi-class convolutional neural networks [12]. They proposed frequency-based filtering and concatenation of the part-of-speech of the word with its embeddings. Sboev et al. suggested a convolutional neural network with long-short term memory cells for author identification in social media documents [8]. Ruder et al. compared the author identification performances of different convolutional neural network models on web collections. They showed that one-hot vector of characters is more effective than non-topic sensitive word embeddings on short texts [86]. Shrestha proposed vectors of character n-grams in a convolutional deep neural network utilized for author identification in short documents [123]. Brocardo et al. experimented with pre-

training deep learning layers which consist of restricted Boltzman machines for author verification. They showed significant improvements on email and twitter datasets [124]. Safder et al. proposed a recurrent model constructed by long-short time memory cells for extracting stylometric features [13]. They used frequent words in the input of the recurrent model and exploit the recurrent output for author verification of literary documents. On the other hand, topical and lexical properties of text have been effectively utilized in joint learning of deep learning models [125].

Although DL approaches have been successfully applied in AA tasks, their performances are dependent on the number of samples. For small number of samples, the domain knowledge is incorporated for learning the representations of documents, sentences, and terms [126–129]. Representations extracted via neural language models have been successfully applied in author identification tasks [30, 116, 122, 125, 130]. These representations capture lexical, syntactic, semantic, and discourse level features from plain text and can be feed into any classification method [122, 130–132]. In AA, the document embeddings extracted by neural language models are proposed as an alternative to the complex language processing and feature extraction methods. Document embeddings have been shown to be robust in cross-topic tasks. Gómez-Adorno et al. used concatenation for combining document vectors extracted from n-grams, words, and part of speech tags [30]. They achieved significant improvements in newspaper articles. Ge and Sun trained a neural language model for each user separately for nearest neighbor classification [119]. Kocher and Savoy used nearest neighbor search for document embeddings extracted by a neural language model [133]. Agun and Yilmazel used document embeddings for the classifier input in an author identification dataset [130].

2.7. Size Issues

There have been numerous studies for measuring the effects of size in AA. The studies regarding the size can be grouped into three categories based on the document size, the number of authors, and the number of documents. Document size is typically referred either to the total number of characters or to the total number of words. An evaluation has shown that the document size should be at least 2500 words to acquire a stable AA prediction performance in novels [71]. However, AA performance on very short documents has been also shown to be successful in different studies [20, 73, 134].

Such studies have been done with a large number of documents per author rather than large document sizes. However, increased number of documents may increase the variability of topics and the noise, affecting the performance of AA [108]. Moreover, increased number of documents creates efficiency problems in classification systems. Such scalability issues are analyzed in detail by Luyckx [135]. On the other hand, the performance degradation in datasets with a large number of authors has been regarded as one of the major problems of AA systems by several studies [136–139]. Lyckx has reported a consistent performance behavior for the number of authors ranging from 10 to 50 in such a way that when the feature overlap size between authors is increasing, the performance is also increasing [140]. Beyond these observations, the binary prediction based approaches have been reported for datasets with large number of authors to be better than the multiclass classification approaches [139, 141].

2.8. Scalability

Scalability problems in AA have emerged after the applications on social media documents with a large number of candidate authors. Scalability problems arise in tasks where training and inference become a bottleneck due to the constraints of storage and processing. Although both author identification and author verification suffer from the scalability issues, the main problem arises in inference for author verification. In the verification tasks unlike author identification, there are many candidate authors and documents. During inference, a document should be tested for each author. Therefore, the author verification brings different scalability problems, especially when a document is assigned to an author among a thousand candidate authors. Either to verify authors or to increase accuracy, the similarity methods on effective feature sets have been an example of scalability in AA [78]. Burrow's delta methods, compression models and nearest neighbor classifier have been widely experimented [24, 142, 143]. Burrow's delta is a similarity metric for measuring the likelihood of an unknown/disputed document written by a given author. It uses z-score computation of a document by a given author profile [144]. Similarly, compression methods use the author's compression dictionary to obtain a compression score of an unknown/disputed document. Later, this score is used in nearest neighbor classification [145]. Common N-Grams (CNG) and several different feature selection techniques are applied in order to diminish the effective size of document vector in nearest neighbor classification [108]. Also, the evaluations on a large number of

document collections use PCA for both dimensionality reduction and nearest neighbor classification. However, the appropriateness of the PCA on a large number of document collections has been questioned in several studies [140, 142].

2.9. Compression Methods

Compression methods are among a few interesting techniques applied in AA tasks. Compression models use compression distance in order to measure the dissimilarity of two documents [141]. They work on the assumption that two documents can be compressed by each other's dictionary easily if they are written by the same author. Compression dissimilarity of intra (same author) and inter (different authors) authors is used in nearest neighbor assignment of an unknown/disputed document [146]. Lambers and Veenman proposed a prototype model for compression measures [146]. Li increased the number of versions of the prototypes via bootstrap sampling of the paragraphs inside the document [147].

2.10. Datasets and Approaches

AA studies have been applied on diverse sets of datasets. These datasets have different properties in terms of topicality, document character length, the type of source and genre. Among some of the popular datasets, Enron dataset is the most widely studied dataset, which consists of email documents of varying topics and lengths [148]. A sampled version of this dataset has been experimented in PAN authorship identification competitions in 2011. In PAN authorship, attribution competitions C10 and C50 are also used. These datasets are extracted from Reuters newspaper corpus [149]. All of the PAN datasets have balanced author and topic distributions.

Recent authorship analysis has been done on documents collected from the web. Although the techniques applied on the web documents have been very similar, the diverse nature of the text length, genre and topicality on different type of documents yield varying results. The types of web documents used in AA have been sampled from reviews [139, 150], micro blogs [150–153], blogs [20, 150, 154, 155], instant messages [4, 138, 156], newspaper articles [5, 157, 158], and comments [4, 20, 159]. On the other hand, studies about authorship analysis of disputed documents have primarily focused on historical documents, such as letters [53, 160], plays [55, 161], literary [50, 162] and religious texts [163]. Early disputed authorship studies applied multivariate analysis

techniques with readability measures [54, 163], vocabulary richness and function words on these documents. With the applicability of machine learning models in disputed authorship, Bayesian methods [54, 164] and feed forward neural network models [81] have been suggested.

Applications of forensic investigations on the web documents have originated new datasets for the AA tasks. Several datasets have been proposed for cross-domain, cross-topic, cross-genre, multi-topic, stylometric clustering, and style breach detection. Cross tasks include a dataset where training and testing document sets are sampled from two different sets. For instance, in cross-domain tasks, the training documents sampled from product reviews and testing documents sampled from technology news are examples of cross-domain dataset. A cross-domain and multi-topic dataset have been proposed for PAN 2018 competitions [165]. Sapkota et al. proposed structural correspondence learning – a machine learning approach – for improving the performance of authorship identification in a cross-domain and multi-topic datasets [88]. Multi-topic datasets have been widely used in AA studies. Halvani et al. compared the performance of AA in multi-topic datasets compiled from essays, news articles, novels, and comments for the languages of Dutch, English, Greek, Spanish and German [68]. Methodologies designed for cross-topic tasks are limited. Markov suggested character n-grams and feature tuning in cross-topic author identification [28]. Stamatatos suggested text distortion on terms to improve performance in cross-topic tasks [22]. Sapkota et al. showed that using the documents of an author in different topics for classifier training is so useful for cross-topic author identification [9]. Castro et al. suggested different similarity measures on linguistic features for cross-topic and cross-genre author verification tasks [94]. Cross-genre tasks involve training and testing documents of an author in different genres. Studies suggested that cross-genre tasks are harder than cross-topic tasks [68, 69, 166]. Halvani et al. claims that common linguistic features, such as morphemes or suffixes in morphologically rich languages help to improve the performance in cross-genre tasks [68]. Kestemont et al. proposed a feature selection strategy known as “unmasking” for cross-genre tasks [69]. They showed a significant performance improvement for five contemporary authors who write in theatre and pose genres.

2.11. Topicality and Approaches

Topical influence on the writing style has been one of the major issues in stylometric analysis tasks. Pioneer research has been conducted on Federalist papers by Mosteller and Wallace in 1984 [167]. They suggested function words as a topic-neutral feature set. Studies on topical influence have focused on the comparison of different feature categories. Function words, character n-grams and punctuations have been suggested to improve the performance of AA in multi-topic and cross-topic tasks [72, 76]. The feature engineering and controlling the sample distributions are mainly investigated in topical influence. Sapkota et al. improved the prediction performance by typed character n-grams for news articles [76]. Mikros and Argiri measured the topic and author correlation of function words, readability and vocabulary richness measures on a controlled corpus compiled from the topics of politics and culture [10]. They suggested that some of these stylometric features are also correlated with topics, and their usage in AA tasks should be questioned. Sapkota et al. showed that mixing the topics of an author improves the prediction performance in cross-topic tasks [9]. Similarly, Schein et al. proposed a validation approach to tune the machine learning model on mixed topics [33]. The common n-grams method – a feature selection method – has been successfully applied on Wikipedia articles [108]. Unmasking is proposed to find pivot variables for selecting features by a topic controlled dataset [168]. Björklund and Zechner have proposed a syntactic feature set to achieve robust prediction performance in a multi-topic dataset compiled from novels [66]. Yang et al. proposed a generative approach for joint inference of topics and authors [169]. They applied a generative model to the documents to extract a latent vector and compared each author's documents with this vector in nearest neighbor classification. Stamatatos achieved significant improvements by applying text distortion in multi-topic and cross-topic tasks [22].

2.12. Scaling and Normalization

In most of the text classification approaches, the importance of document processing has been specified [28, 99]. Among such processing techniques, term weighting, normalization and document vector scaling are widely applied. These techniques are also available to authorship attribution tasks. Normalization has been used for combining different sets of features [15]. Discretized document vectors have been

reported to be useful for large scale AA [79, 170]. Such scaling techniques as mean, min-max and z-score have been successful in different AA datasets [63, 113, 156, 171].

Tanguy et al. applied discretization and feature normalization for each set of feature group separately by using max and mean normalization [63]. They reported a significant improvement with the feature discretization and a slight improvement with the mean and max normalization. Savoy observed a minor performance difference in applying mean scaling prior to PCA [113]. Goldstein-Stewart applied averaging on the term features in order to select useful terms [156]. On the other hand, z-score normalization has been widely applied in similarity based approaches. Evert and Savoy applied z-score normalization for Delta measures [143] and KLD distance schema [172] respectively.

3. AUTHORSHIP ATTRIBUTION

The performance of text classification applications depends on the number of categories, the number of samples for each category and the number of features for each sample. In traditional text classification tasks, the categories contain information generated by the human knowledge or expertise. Such categories are related with the document content. For instance, in a topic classification task, each topic can be easily inferred from the words of the document. However, in AA, since the author style is correlated with the context, the extraction of the author style is much harder than topic extraction. Thus, AA problems are generally referred as the needle in the haystack problem. Moreover, classification features are not similar to text classification applications. Function words, which are an important feature for AA, are removed in topic categorization. These properties make AA harder than text classification applications. In the following sub-sections, the details of the general methodologies applied in AA tasks are presented.

3.1. Learning from Document Representation

In machine learning applications, the instances are represented in vector form. Features derived from text are further processed for the vectorization in order to be used directly in inference and classification tasks. The text is represented in vector form on the assumption that all the vector elements are independent and identically distributed. Although the information represented by the order of words in vector representation are lost, an example of such representation known as bag of words (BOW) has been widely applied and become the most successful approach in text categorization and authorship attribution. In BOW, each document is represented by a feature vector where elements are the dictionary items extracted from text and values are their frequency in the document. In most AA approaches, BOW representation is used for storing not only the information about text elements, such as terms, characters, named entities and the like. but also statistical information, such as document lengths in characters, distinct number of terms, number of spaces per tokens and the like..

In BOW representation, each feature is added to the dictionary. Later, the dictionary is used to construct document vectors. Each item in the dictionary constructs an element for the index of the item for the document vector. Item frequency, or presence (1) and absence (0) in the document are used as values of elements of the document vector. For

extraction of dictionary items, such as terms, punctuations, dates, named entities, symbols, function words, character n-grams, collocations and other forms of text elements, the language processing by tokenization, n-gram extractor, regular expression matching, sentence detection, named entity recognition and etc. is applied on the text. An example schema for constructing dictionary and processing document is given in Figure 3.1 and is expanded in Figure 3.2

.

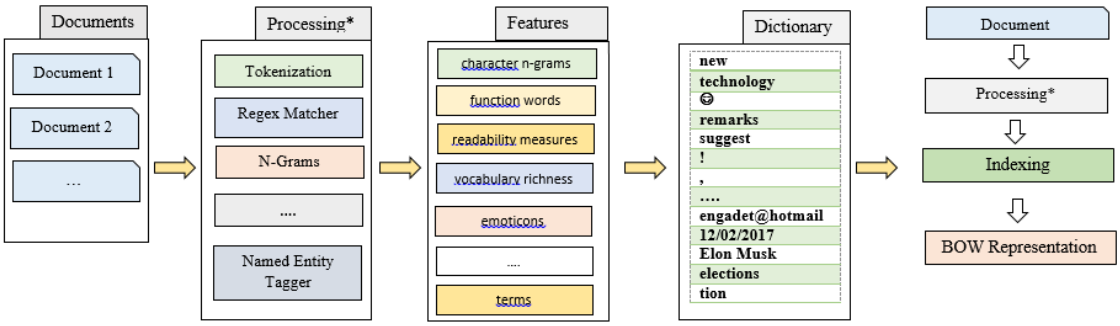


Figure 3.1. *Document processing*

In Figure 3.1, all the documents are processed in order to build a unified dictionary. For each processing step on all the documents, the distinct set of items is added to the dictionary. Each dictionary element in the unified dictionary is distinct for a given processing task. After the dictionary is constructed, final BOW representation is obtained by indexing. Indexing stores the statistical information of the processed item at the position determined by the index of the item in the dictionary so that each item is represented by a value at its index position. The general pipeline for obtaining a document vector may involve a weighting and selection stage where each element in the BOW representation is weighed and either eliminated or transformed during the construction of the final document vector.

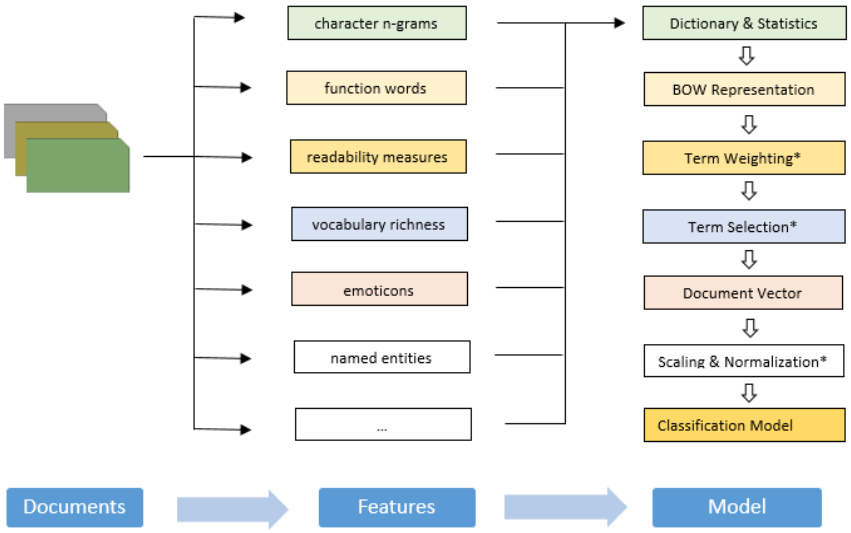


Figure 3.2. *Model Pipeline*

In Figure 3.2, the final steps of constructing the document vector and building the classification model are given. Optional steps, which are term weighting, term selection and scaling/normalization, are marked by asterisk. Term weighting assigns appropriate weights to each element in the BOW representation. These weights are calculated from the statistical properties of dictionary items and applied to term selection for filtering or transforming the elements of BOW representation. For example, principal component analysis – a common dimensionality reduction technique – maps the BOW representation to an orthogonal space and reduces the dimensionality according to the importance of the orthogonal components. On the other hand, term selection schemas select important terms based on the computed scores from term weights. The final steps before building the classification model are scaling and normalization methods. These methods are useful for classifying documents with different lengths and word densities.

3.2. Common Stylometric Features

Common state of art approaches have been successfully applied by modification of existing stylometric features in the early AA tasks. Studies on measuring topic and genre effects on AA suggest that character n-grams, function words and terms have been in correlation with topic and genre. For this reason, modifying or extending existing feature sets has been an important issue in AA literature. In this section, we explain the details of the most useful features in stylometric analysis.

3.2.1. Length ratios

Length ratios are a feature set consisting / which consists of character lengths of paragraphs, sentences, tokens, words, punctuations, digits and spaces. The formulations of this feature set of character lengths for token, word, digit, punctuation and space patterns are given in Table 3.1.

Table 3.1. *Formulations of length ratios*

Feature Name	Equation
total text length	$\sum_{tokens} length(token)$
total word length	$\sum_{words} length(word)$
total punctuation length	$\sum_{punctuations} length(punctuation)$
total digit length	$\sum_{digits} length(digit)$
total space length	$\sum_{spaces} length(space)$
average sentence length	$\sum_{sentences} \frac{length(sentence)}{number\ of\ sentences}$
average paragraph length	$\sum_{paragraphs} \frac{length(paragraph)}{number\ of\ paragraphs}$
average word length	$\frac{total\ word\ length}{number\ of\ words}$
average punctuation length	$\frac{total\ punctuation\ length}{number\ of\ punctuations}$
average space length	$\frac{total\ space\ length}{number\ of\ spaces}$
frequency of tokens above length of 5	$\sum_{tokens} count(length(token) > 5)$
frequency of tokens below length of 10	$\sum_{tokens} count(length(token) < 10)$
ratio of word length	$\frac{total\ word\ length}{total\ text\ length}$
ratio of punctuation length	$\frac{total\ punctuation\ length}{total\ text\ length}$
ratio of digit length	$\frac{total\ digit\ length}{total\ text\ length}$

In Table 3.1, we used total, average and frequency based measures in order to represent the ratios in this feature set. The count function in frequency-based measures is a test function that returns one for true and zero otherwise. The examples of these measures are given in Table 3.2.

Table 3.2. *Examples of basic measures*

Example Text	Length Type	Result
retrieval of language ..	total token length	21
retrieval of language ..	number of tokens	4
retrieval of language ..	total word length	19
retrieval of language ..	number of words	3
in 11/07/2017 a very pretty girl	total digit length	8
in 11/07/2017 a very pretty girl	number of digits	3
it's during the world war …	total punctuation length	4
it's during the world war …	number of punctuations	2
the° most° ° ° valuable° asset	total space length	5
the° most° ° ° valuable° asset	number of spaces	3
the most valuable asset	frequency of tokens above length of 5	1
the most valuable asset	frequency of tokens below length of 10	4

In Table 3.2, there are three types of measures: the total pattern length, the number of patterns and the frequency filter count. The patterns correspond to text fragments, such as token, word, digit, punctuation and space. The total pattern length is the count of characters in a given text fragment and the number of patterns is the count of non-continuous patterns. For instance, the total space length is computed by summing all the lengths of non-continuous spaces. On the other hand, the frequency-based filter count is applied to count the patterns above or below certain character length.

3.2.2. Length histograms

Word length and word length histograms are one of the first suggested stylometric features [1]. Word length histograms are a frequency-based representation where each

word is grouped by its length, and the number of words in each length is counted. The word length grouping is simply done for the length of words or a custom range of length. For example, the count of words with the length above five and below ten is an example of grouping. Similarly, each word can be grouped by its own length. In Table 3.3, a simple example of representing a document by length histograms is given.

Table 3.3. *Word length histograms*

Dictionary Items	DF	Lengths	Groups	Items In Length	Document Vector
@	0	1	1	@{0}	
			2		0
!!!	0	3	3	!!!{0}	0
Tech	2	4	4	tech{2}, size{1}, ball{1}	0 4
gadget@hotmail	0	14	5		0
			6		0
filtering	0	9	7		0
application	1	11	8	polarity{2}, negative{1}	3 0
Polarity	2	8	9	filtering{0}	0
			10		1
negative	1	8	11	application{1}	0
			12		0
Size	1	4	13		0
Ball	1	4	14	gadget@hotmail{0}	0
			15		

In Table 3.3, the first column is the set of dictionary items for a dataset. A sample document vector is given in the second column, where each element is represented by the frequency of corresponding dictionary item in the document. For all dictionary items, the grouping is done by their length and counts in the document. The grouping column holds the frequencies of these items by the given length index in curly brackets. For example, the term “application” appears one time in the document, and it has a length of eleven, so it is specified in the index of eleven with its frequency. The final document representation is a vector of the total counts for each length. It is constructed by summing frequencies in each group row.

3.2.3. Readability measures

Readability measures are first designed to grade the language comprehension of an author. They are widely used in the prediction of age and gender – a stylometric task known as author profiling [173–175]. In this thesis, we applied the readability measures

in combination to vocabulary richness measures. The applied measures are automated readability index, Coleman-Liau readability formula, Flesch reading ease formula, Flesch-Kincaid grade level, Gunning Fog formula, LIX readability formula and SMOG index. The formulations of these measures are given in Table 3.4. In these formulations, the items of characters, words, periods, vowels and sentences are represented by the number of occurrence. For instance, the sentences are an abbreviation for the number of sentences.

Table 3.4. *Formulations of readability measures*

Readability Measures	**Formulas**
automated readability index	$4.71\ x\ \left(\frac{characters}{words}\right) + 0.5\ x \left(\frac{words}{sentences}\right) - 21.43$
Coleman-Liau readability formula	$0.0588\ x \left(\frac{characters}{words\ mod\ 100}\right) - 0.296\ x \left(\frac{sentences}{sentences\ mod\ 100}\right) - 15.8$
Flesch reading ease formula	$206.835 - \left(1.015\ x\ \left(\frac{words}{sentences}\right)\right) - \left(84.6\ x\ \left(\frac{vowels}{words}\right)\right)$
Flesch-Kincaid grade level	$\left(0.39\ x\ \left(\frac{words}{sentences}\right)\right) + \left(84.6\ x\ \left(\frac{vowels}{words}\right)\right)$
Gunning Fog formula	$0.40\ x\ \left(\frac{words}{sentences} + \frac{long\ words}{number\ of\ words}\right)$
LIX readability formula	$\left(\frac{words}{periods}\right) + \left(\frac{long\ words\ *100}{words}\right)$
SMOG index	$3 + \sqrt{words\ with\ three\ or\ more\ vowels}$

In Table 3.4, the modified or approximated formulations of the listed readability measures, the ones in the first column, are given. For instance, the original formulations of Flesch reading ease, Flesch-Kincaid grade level and SMOG index contain the number of syllables of the words, and we approximated this number by counting the number of vowels for computational efficiency. For Turkish language, replacing syllables with vowels is a good approximation, but for English language, there are certain words in which this replacement simply doesn't work. For this reason, we used a trivial finite state automata approach to count English syllables. The regular expression syntax of this automata is [aeiouy]+[^$e(,.:;!?)]. This automata is used to count the characters which

consist of "a", "e", "i", "o", "u", "y" that is not followed by a punctuation mark or the character "e". In LIX and Gunning Fog formulas, the long words are given by the count of the words with a length of 6 or above.

3.2.4. Vocabulary richness

Vocabulary richness measures have been widely used in stylometric analysis tasks. Since vocabulary richness measures are based on the number of words, they are dependent on the document length [176, 177]. Therefore, the vocabulary richness measures are suggested as a complementary feature set [178, 179]. Eight well-known vocabulary richness measures, which are given in Table 3.5, are used in the experimental evaluations.

Table 3.5. *Formulations of vocabulary richness measures*

Vocabulary Richness Measures	Formulation
Guiraud's R	$\frac{V}{\sqrt{N}}$
Herdan's C	$\frac{log(V)}{log(N)}$
Rubet's K	$\frac{log(V)}{log(log(N))}$
Maas A	$\frac{log(N)/log(V)}{(log(N))^2}$
Dugast's U	$\frac{(log(N))^2}{log(10)\ /log(V)}$
Janenkov-Neistoj	$\frac{1}{((V\text{^}2 * log(N))\ )}$
Brunet's W	$(V\ -\ 0.172) *\ log(N)$
Hapax Dislegomenon	$The\ number\ of\ distinct\ tokens\ appearing\ twice$
Hapax Measure	$\frac{Hapax\ Dislegomenon}{V}$

In Table 3.5, the vocabulary richness measures are computed for each document separately by using the number of distinct terms (V) and the total number of tokens (N) in the document. The Hapax Dislegomenon is computed by counting the number of distinct terms which appear twice in the document.

3.2.5. Terms

Term feature set has been a common feature set in text classification tasks. Even though they are indirectly used through several measures, such as delta measure, word histograms, vocabulary richness and readability measures in AA tasks for a long time, recent author identification approaches showed their effectiveness in multi-topic tasks [13, 28, 60, 158]. The term feature set is extracted by tokenization - a text-processing task applied to determine the boundaries of meaningful segments, such as words, dates, names, symbols and punctuations. In Figure 3.3, a simple example for tokenization is given.

Sentence:	Beşiktaş: `Topu hücum sahasına taşımak gerekiyor.`									
Tokens:	Beşiktaş	:	`	Topu	hücum	sahasına	taşımak	gerekiyor	.	`
Lowercase:	beşiktaş	:	`	topu	hücüm	sahasına	taşımak	gerekiyor	.	`

Figure 3.3. *Tokenization example*

In Figure 3.3, after tokenization, we apply text standardization for each token. By using text standardization, each token is converted to its lowercase. The lowered tokens are chosen as features and used in construction of the term vocabulary.

3.2.6. Function words

Function words are among the most notable stylometric features in AA studies [75, 78]. In numerous studies, the distinction between function words and content words has been properly drawn [67]. Their usage in text has been claimed to be an effective analysis tool for human cognition and behavior. Chunk has noted that the differences of function words used by three people in an ice cream recipe determines their linguistic style [74]. The function words are used for linking other words. Thus, their purpose are not

dependent on the language, but their surface forms are simply different for different languages. Function words include auxiliary verbs, adverbs, determiners, conjunctions, pronouns and prepositions. In this thesis, the general function word extraction methodology is demonstrated in Figure 3.4.

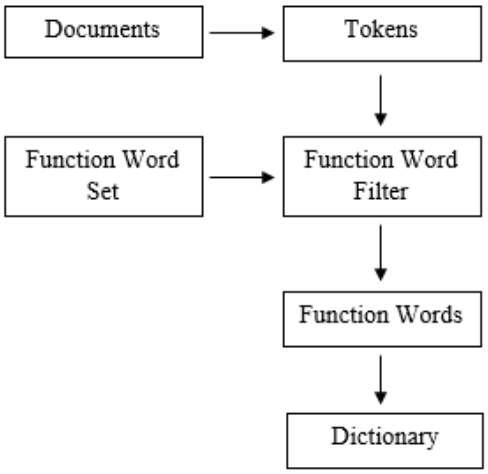


Figure 3.4. *Steps of function word filtering*

In Figure 3.4, the documents are tokenized, and function words are filtered among these tokens by using a predefined set of function words. The statistical properties of function words are stored in the dictionary, and later, the statistics are used in TF-IDF term weighting schema. Although this is the common function word extraction methodology, some morphemes in Turkish language should be considered as function words. In a machine translation study, mapping the function words to the corresponding morphemes has been referred as a common problem [180]. The demonstration of different morphemes in Turkish is given by morpheme segmentation studies [181].

Table 3.6. *Example for function word behavior of Turkish morphemes*

Category	Turkish	English	Function Word
Word	Düşünemediklerinden	since they couldn't think	No
Verb	Düşün	think	No
Ability+Negative	Eme	couldn't	Yes
Past	Dik	could	Yes
Thrid Person Plural	Leri	they	Yes
Noun	-	-	No
Ablative	nden	since	Yes

In Table 3.6, an English translation for a Turkish word where some of the Turkish surface forms behave like function words is given. Each column in the table is dedicated to morphological categories, translations and function word behavior of Turkish and English surface forms. According to the translation of the Turkish word, the demonstration of the surface forms contains three distinct function words, which are "could", "they" and "since". For this reason, in Turkish language, a morphological analyzer should be utilized for extracting function words by filtering certain types of surface forms [99]. However, in our experimental evaluations, the traditional function word extraction methodology is applied for simplicity.

3.2.7. N-grams

N-gram features have been widely applied in text classification tasks. There are two different versions of n-grams. These are continuous n-grams and skip n-grams. Continuous n-grams extract adjacent text segments, and skip n-grams extract both continuous and discontinuous text segments. Both types of n-grams are suggested to capture dependencies in the text segments where bag-of-words representation fails to capture because of the independence assumptions. N-grams can be applied on text boundaries, such as tokens, characters, syllables and the like. In the experimental evaluations, we apply continuous n-grams on characters. In Figure 3.5 and Figure 3.6, it is demonstrated how word n-grams and character n-grams are extracted by the moving window approach.

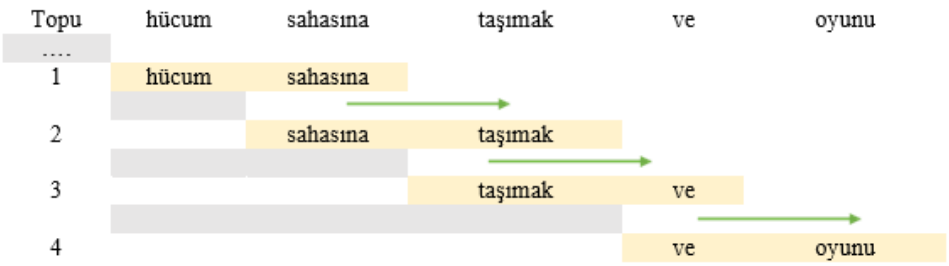


Figure 3.5. *Moving window for word n-grams*

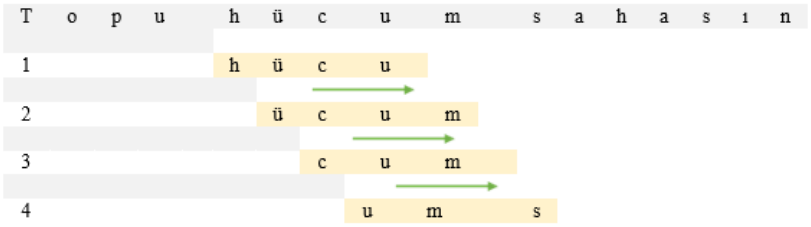


Figure 3.6. *Moving window for character n-grams*

In Figure 3.5 and Figure 3.6, the moving window approach is applied to extract two-word length and four-character length text segments respectively. In Figure 3.5, the window size is two, and the extracted word sequences are composed of two consecutive words. In Figure 3.6, the window size is four, and the extracted character is composed of four consecutive characters.

Although traditional n-grams extraction is employed on the whole text of the document, there are several approaches that employ n-gram extraction on different parts of text. For instance, typed n-grams have been shown to be more successful than traditional n-grams in several AA tasks. The extraction process of typed-ngrams is different than traditional n-grams. Typed n-grams use word suffixes, prefixes as well as word boundaries and combine this information to the extracted n-grams with various lengths. For example, the word “sahasına” has prefix n-grams and suffix n-grams. Prefix n-grams with a minimum length three contain the character n-grams of “P=sah” and “P=saha”, and suffix n-grams with minimum length of three contain the character n-grams of “S=ına” and “S=sına”. In these examples, P and S correspond to prefix and suffix types respectively.

3.3. Common Applied Methods

The document frequency (DF) and TF-IDF term weighting are common techniques applied for selecting or filtering features. Document frequency-based filtering and common n-grams (CNG) have improved the performance in most of the AA datasets. Delta rule has been a pioneer statistical analysis method in AA tasks to remove redundancies in each feature. Along with Delta rule, multivariate analysis techniques, such as principal component analysis (PCA) have been widely applied to visualize and analyze stylometric patterns in AA. Normalization and scaling are also important techniques, which have been applied in classification of heterogeneous documents. In the following subsections, each of these methods is explained in detail.

3.3.1. Document frequency (DF)

Document frequency (DF) is the occurrence rate of a feature among the documents. DF is a simple and effective filtering and analysis approach, based on the assumption that discriminative features have similar document frequencies. For example, in topic classification, eliminating very common and rare features with high document and low frequencies respectively is reported to be an effective feature selection approach. On the other hand, features with high document frequency have been reported to be effective in AA. A DF filtering example is given in Table 3.7.

Table 3.7. *Document Frequency Filtering*

Dictionary Features	**Document Frequencies**	**Document Frequency Filters**		**Filtered Set**	
		>= 5	**>= 8**	**>=5**	**>=8**
@	12	1	1		
!!!	2	0	0		
Tech	24	1	1	@	@
gadget@hotmail	2	0	0	tech	tech
Filtering	5	1	0	filtering	
Application	7	1	0	application	
Polarity	2	0	0		

In Table 3.7, the document frequency-filtering example is demonstrated for seven dictionary terms with different document frequencies. The terms with document frequencies above or equal to five and eight are filtered in the table respectively. The filtered terms are given in the last column in the table, where the filter of five contains all the terms of the filter of eight.

3.3.2. Common n-grams (CNG)

Author-feature correlation is an important property of stylometric analysis. Common n-grams are suggested to find topic independent n-gram features. CNG have two common variants, namely local CNG [182, 183] and global CNG [107–109]. Local CNG filters the n-grams by using the total term frequency for all the documents for a given author. Later, the filtered n-grams for each author are unified. On the other hand, global CNG filters n-grams according to their term frequency among all documents. CNG with variants is a simple yet the most successful frequency-based term selection technique applied to AA. An example for CNG filtering is demonstrated in Table 3.8.

Table 3.8. *Local CNG and Global CNG Filtering*

Authors	**Term Frequency of N-Grams**				**Filtering by Local CNG**	**Filtering by Global CNG**
	Document 1					
A1	İng	20	…	20	ing ion com … bat reg ncy thi	ing ion hav com !!! … bat reg ncy thi suc
	İon	25	Bat	3		
	Hav	13	Reg	18		
	Com	0	Ncy	13		
	Sur	0	Thi	20		
	!!!	0	Suc	1		
	Document 2					
A1	ing	27	…	12		
	ion	12	Bat	5		
	hav	10	Reg	8		
	com	1	Ncy	13		
	sur	1	Thi	2		
	!!!	2	Suc	2		
	Document 3					
A2	İng	4	…	2		
	İon	5	bat	15		
	Hav	3	reg	18		
	Com	20	ncy	3		
	Sur	10	Thi	10		
	!!!	16	suc	15		
	Document 4					
A2	ing	3	…	18		
	ion	2	Bat	12		
	hav	3	Reg	1		
	com	12	Ncy	1		
	sur	7	Thi	17		
	!!!	8	Suc	8		

In Table 3.8, CNG filtering example with frequency above 25 is given for two authors and four documents. Each document and author is presented in the first two columns. Documents are represented by the frequency of the items in the document, where if the document doesn't contain the item then the frequency is marked as zero. The

authors along with the document identifiers are given for a hypothetical example. N-Grams of size 3 and their term frequencies in each document are given. A filtering stage eliminates frequencies below 25. While global CNG eliminates only "sur" and keeps the rest, local CNG simply eliminates "hav", "sur", "!!!", and "suc". Local CNG sums up frequencies in all the documents of an author, and elimination is done for each author separately. After elimination, each result is unified into single dictionary. On the other hand, elimination in global CNG is done similarly but once for all documents.

Both CNG methods are using term frequency in filter based term selection. However, document frequency can be used for selection. Using document frequency in global CNG is the same as DF filtering. Nevertheless, local CNG significantly differs from DF filtering. For this reason, a term weighting methodology based on local CNG for imbalanced datasets is suggested. CNG term weighting for a given term is given in Eq. (3.1).

$$weight(term_i) = 1/M \sum_{k}^{M} \frac{DF\ of\ term_i\ in\ documents\ of\ author_k}{Document\ count\ of\ \ author_k} \tag{3.1}$$

In Eq. (3.1), document frequency (DF) of $term_i$ in $author_k$ is the number of documents of $author_k$ where $term_i$ appears and document count of $author_k$ is the total number of documents written by $author_k$. M represents the number of authors in the document collection. After local CNG weighting is applied, terms with higher weights can be selected. For instance, in order to filter the most important terms, selecting terms with top 50 highest weights can be applied to reduce the size of dimensionality to 50.

3.3.3. TF-IDF term weighting

Term frequency and document frequency are two important statistics of a feature in BOW representation. Term frequency is the count of a feature in a given document, and document frequency is the count of presence of a feature among all the documents. Inverted document frequency (IDF) assumes that common features have low discriminative power, and rare features are discriminative. TF-IDF ensures the balance between local and global importance of a feature. For most of the AA datasets, the reliability and effectiveness of TF-IDF weighting has been proven [184].

Computation of TF – IDF vector is in two folds: Inverted document frequency (IDF) calculation for each dictionary term and weight calculation by term frequency (TF)

multiplication. IDF for a term is computed as in Eq. (3.2) and the weight of term "t" is calculated by multiplying IDF and TF as in Eq. (3.3).

$$idf_t = log\left(\frac{N+1}{DF_t+1}\right) \tag{3.2}$$

$$term_t = tf_t \times idf_t \tag{3.3}$$

In Eq. (3.2), DF_t is the number of documents that the term "t" occurs; 1 is added for smoothing, and log is the logarithm in base 10. An example of document vector construction from two documents is given in Table 3.9 and Table 3.10.

Table 3.9. *Term weighting and document vectors*

Documents	Tokens	TF-Vector	Document Vector
If everything is going right ... something is wrong.	if, everything, is, going, right, …., something, is, wrong, .	[1, 1, 2, 1, 1, 1, 1, 1, 1, 0, …, 0]	[0.0, 0.176, 0, 0, 0.176, 0.176, 0.0, …, 0.0]
He is going to do something if it is wrong.	he, is, going, to, do, something, if, it, is, wrong	[1, 0, 2, 1, 0, 0, 1, 1, 1, 1, …, 1]	[0.0, …, 0.0, 0.176, 0.176, 0.176, 0.176]

Table 3.10. *Dictionary and statistics*

Items	Vectors
Dictionary	[if, everything, is, going, right, …, something, wrong, ., he, to, do, it]
DF Vector	[2, 1, 2, 2, 1, 1, 2, 2, 2, 1, 1, 1, 1]
IDF Vector	[0.0, 0.176, 0.0, 0.0, 0.176, 0.176, 0, 0, 0, 0.176, 0.176, 0.176, 0.176]

Table 3.9 contains every processing stage from left to right. In Table 3.10, corresponding DF and IDF vectors are constructed from each dictionary term and their document frequency. Final document vector is given in the last column of Table 3.9. It is obtained by element-wise multiplication of TF-vector and IDF vector.

3.3.4. Burrow's Delta

Burrow's delta is a measurement technique based on the z-score deviation in the frequency of each feature for a given author and the dataset [172, 185]. Delta is computed for each document separately. Several different versions of delta measures have been

suggested for AA tasks [186, 187]. All suggested delta measures use a distance function between the given document and the compared author profile for the nearest neighbor classification [143]. The common application of delta measures uses feature selection or controlled vocabulary and normalization prior to z-score computation. The general z-score calculation of ith term in the vocabulary is given by Eq (3.4).

$$z - score_i(doc) = \frac{freq_i(doc) - mean_i}{standard\ deviation_i} \tag{3.4}$$

In Eq (3.4), the z-score is computed for a given document where $freq_i(doc)$ is the frequency of the ith term in the document, and $mean_i$ and $standard\ deviation_i$ are the mean and standard deviations for the frequency of the ith term in the dataset. After z-score is computed for each document, author profiles are extracted for each author by averaging the corresponding documents z-score values.

3.3.5. Principal component analysis

Principal component analysis (PCA) is a multivariate analysis technique based on orthogonal decomposition. It has been widely applied for dimensionality reduction in text classification tasks. PCA was a pioneer approach for analysis of decision boundaries for authors in the early AA studies [54, 55]. In recent AA tasks, it has been widely applied for feature selection and transformation [4, 58, 188]. PCA is generally defined as an orthogonal transformation method based on maximization of the variance in an orthogonal coordinate system. The general notation for the first principal component is given in Eq. (3.5).

$$w_1 = argmax\{\frac{w^T X^T X w}{w^T w}\} \tag{3.5}$$

In this equation, w_1 is the first principal component, and X is the data matrix where the rows correspond to the document vectors. The corresponding w that satisfies this criterion is the first eigenvector for the largest eigenvalue in eigendecomposition of $X^T X$. The kth principal components can be found by subtracting the first k-1 principal components from X. These steps are given in Eq. (3.6) and Eq. (3.7).

$$X_k = X - \sum_{s}^{k-1} X w_s w_s^T \tag{3.6}$$

$$w_k = argmax\left\{\frac{w^T X_k^T X_k w}{w^T w}\right\} \tag{3.7}$$

It is also shown that the next principal component is the eigenvector of next largest eigenvalue in the eigendecomposition of $X^T X$. After the eigendecomposition is applied, all the principal components are extracted, and the final transformation matrix is formed from the eigenvectors with the kth largest eigenvalues. The final transformation matrix is applied to the original data matrix as given in Eq. (3.8).

$$T = XW \tag{3.8}$$

The number of principal components is selected by using experimental evaluation. In this evaluation, the goal is to find the minimum number of principal components so that the %99 of the variance in the original data points is preserved. The criteria that satisfied this condition is given in Eq. (3.9).

$$1 - \frac{\sum_{i=1}^{k}(\lambda_i)}{\sum_{i=1}^{n}(\lambda_i)} \leq 0.01 \tag{3.9}$$

In this equation, λ_i is the ith eigenvalue, n is the number of non-zero eigenvalues, and k is the number of top largest eigenvalues in the eigendecomposition of XX^T.

3.3.6. Feature normalization

In text classification tasks, normalization and scaling methods have been successfully applied to standardize the document vectors due to varying genre, topic, length and source of the documents. Normalization is commonly applied in classification tasks where the frequency of features diverges across different document lengths. Normalization is also known as feature scaling where the features are standardized by minimum, maximum values or the norm of the document vector. There are a variety of feature scaling techniques, such as min-max scaling, standard scaling, soft-max scaling, z-score normalization, and mean scaling. These scaling techniques assume that the features are independent and identically distributed. In Table 3.11, the formulations for popular scaling methods are given.

Table 3.11. *Scaling formulations*

Scaling method	Formulation
Standard scaling	$x_i = \frac{x_i - mean(x)}{stdev(x)}$
Min-max scaling	$x_i = \frac{x_i - \min(x)}{\max(x) - \min(\mathrm{x})}$
Euclidean normalization	$x_i = \frac{x_i}{\|\|d_i\|\|}$

In Table 3.11, the formulations are given for the ith document vector (d_i) and the feature "x". Mean, stdev, min, max, are the notation of the mean, standard deviation, minimum and maximum of the given feature in the whole dataset.

3.4. Classification Methods

Instance-based supervised learning has been widely applied to AA tasks [14, 81, 136, 189]. Supervised learning is done in two steps: optimizing parameters for known document vectors and using these parameters for classification of an unknown document vector. The first step is known as training, where each document vector and author pair is used for adjusting parameters to minimize a global loss function. The loss function is used for computation of the error rate. Each type of classification algorithm has different internal optimization procedure to minimize the error. In general, the classifiers are grouped into two types: generative and discriminative. Generative classifiers fit the document vectors with an aim to produce the most likely outcome for a given label, while discriminative classifiers seek the optimum boundaries between the instances of different labels. A general comparison of generative and discriminative has showed that the discriminative approach minimizes the error slower than the generative approach, but it has a lower minimum error rate than the generative approach [190]. In our experimental evaluations, we applied naïve Bayes, logistic regression, decision tree, random forest and multilayer perceptron classifiers, the details of which are given in the following subsections.

3.4.1. Naïve Bayes classifier

Naïve Bayes is the common probabilistic classification approach in text classification applications. It is a simple generative classification model and based on the joint probability estimation of class likelihood from each feature probability with independence assumption. In our experiments, multinomial Naïve Bayes classifier was used. Let $D_i = (x_1, x_2, \dots, x_N)$ represents the ith document vector in the dataset, $A = \{A_1, A_2, \dots, A_K\}$ represents the set of labels then the probability of a document D_i belonging to label A_k is given in Eq. (3.10).

$$p(D_i|A_k) = log(p(A_k) + \sum_{i=1}^{N} x_i \cdot log(p(x_i|A_k) \tag{3.10}$$

$P(A_k)$ is known as class prior probability, and the probability $p(x_i|A_k)$ is estimated in Eq. (3.11), in which F_{ik} is the frequency of the ith feature in all training documents belonging to label k.

$$p(x_i|A_k) = 1 + \frac{F_{ik}}{N + \sum_{i=1}^{N} F_{ik}} \tag{3.11}$$

The decision function of naïve Bayes classifier for an unknown document vector x is given in Eq. (3.12).

$$y = \underset{k\epsilon\{1,..,K\}}{arg\,max}\left(log\big(p(A_k)\big) + \sum_{i=1}^{N} log(p(x_i|A_k) \right) \tag{3.12}$$

3.4.2. Logistic regression classifier

Logistic regression classifier is a linear discriminative classification model. In our study, multinomial (softmax) logistic regression was used. $Let\ D_i = (x_1, x_2, \dots, x_N)$ is the ith document vector in the dataset, $A = \{A_1, A_2, \dots, A_K\}$ is the set of labels and w_0, $w_{1..N}$ are the classifier parameters then the probability of a document D_i belonging to label A_k is given in Eq (3.13).

$$p(y = A_k|D_i) = \frac{exp(w_{k0} + \sum_{i=1}^{N} w_{ki}\, x_i)}{1 + \sum_{j=1}^{K} exp(w_{j0} + \sum_{i=1}^{N} w_{ji} x_i))} \tag{3.13}$$

The weighted negative log-likelihood objective function of logistic regression classifier is given in Eq. (3.14) where, $\big|\big|w\big|\big|_2^2$ is the L2 norm and $\big|\big|w\big|\big|_1$ is the L1 norm of the weights.

$$\min_{w,w_0}\left(-\sum_{i=1}^{N}\omega_j\, log(p(y=y_i|D_i)+\lambda\left[\frac{1}{2}(1-\alpha)\left|\left|w\right|\right|_2^2+\alpha\left|\left|w\right|\right|_1\right]\right) \tag{3.14}$$

3.4.3. Decision tree classifier

Decision tree is a nonlinear classifier and commonly used for building expert systems in machine learning applications. Decision trees are useful for experts in decision-making process thanks to showing decision boundaries and structure of a classification system. A decision tree builds a tree structure to support a decision for each label in each leaf of the tree by traversing the feature values from top to bottom. Decision tree learning builds the tree by recursive partitioning of the feature space at each level of the tree. Each tree node is split according to split objective determined by the information gain of partitioning the node [191–193]. The information gain objective is given in Eq. (3.15).

$$\underset{x_i}{argmax} IG(D,x_i) \tag{3.15}$$

In Eq. (3.15), D is the whole dataset, x_i is ith feature, and IG is the information gain function. It is the difference between parent node impurity and sum of two child node impurities. Impurity at a node represents the measure of the homogeneity of the labels at that node. Formulation of information gain and impurity at a specific node are given in Eq. (3.16) and Eq. (3.17) respectively.

$$IG(D,x_i) = Impurity(D) - \frac{N_{left}}{N} Impurity\left(D_{left}\right) - \frac{N_{right}}{N} Impruity\left(D_{right}\right). \tag{3.16}$$

$$Impurity(D) = \sum_{i=1}^{K} f_i(1-f_i) \tag{3.17}$$

In equations (3.16) and (3.17), K represents the number labels, and f_i represents the frequency of label I at node D.

3.4.4. Random forest classifier

Random forests are generated by ensembles of decision tree models. Training of a random forest is done through bagging. In bagging, training dataset is repeatedly sampled with replacement, and a decision tree model is created from these samples. For an unseen

sample, the prediction is done by the majority voting technique, where all models are evaluated and the class label which has the majority of votes is decided to be the final class label. Increasing the number of tree models prevents overfitting caused by decision tree learning [194].

3.4.5. Multilayer perceptron classifier

Multilayer perceptron classifier is a feedforward artificial neural network [195]. It is a non-linear classifier. A multilayer perceptron classifier is represented by input layer, hidden layers and output layer, through which an input vector propagates. Hidden layers and output layer contain nodes with weights and activation function. Weights of a node are used for processing the outputs of a previous layer. Activation functions are applied to the output of a node. In general, hidden layer activation functions are sigmoid, and output layer activation functions are softmax. The equations for these activation functions are given by Eq. (3.18) and Eq. (3.19) respectively. Input vector propagates through input, hidden and output layers. Error in the output layer given by Eq. (3.13) is back propagated from the output layer to the hidden layers. At each node, the error function is calculated and weights of each node are updated based on the error of the output of that node. Update equation is given by Eq. (3.22), where γ is the step size, $\delta E \,/\delta w_i$ is error derivation with respect to the weights of ith node (w_i).

$$f(z_i) = \frac{1}{1 + e^{z_i}} \tag{3.18}$$

$$f(z_i) = \frac{e^{z_i}}{\sum_{k=1}^{N} e^{z_k}} \tag{3.19}$$

In Eq. (3.18) and Eq. (3.19), the z_i calculated by Eq. (3.20), where w_{ji} is the weight connects j_{th} node output (x_j) to i_{th} node.

$$z_i = \sum_{j=1}^{M} w_{ji}\, x_j \tag{3.20}$$

In Eq. (3.21) and Eq. (3.20), where X_p is the pth input vector to the classifier, W is the weights of the neural network, $\check{y}_{kp}$ is the pth output value for pth input at output layer node k, and y_k is the correct output value in output layer k for the given input, the cross entropy error function for the output layer is given.

$$E(\vec{X}_p, W) = \sum_{k=1}^{N} (\breve{y}_{kp}\, log(\breve{y}_{kp}) - (1 - y_k)\, log(1 - \breve{y}_{kp})) \tag{3.21}$$

$$w_i = w_i - \gamma \frac{\delta E}{\delta w_i} \tag{3.22}$$

3.5. Evaluation Strategies

In a classification task, the model generated from the samples is evaluated on a separate sample collection. The training collection is known as training set, and the evaluation collection is known as testing set. For the optimum performance on the test dataset, the best performing parameters are selected from the parameters that satisfy the minimum error rate on a validation dataset, which is a disjoint set obtained by partitioning the training dataset. Training dataset and testing datasets are later used for training and evaluation respectively.

In order to separate a whole dataset into training, validation and testing collections, random sampling techniques, namely cross-validation and stratified cross-validation, are applied. Each of these techniques partitions the dataset into k disjoint training and testing/validation collections, where the training collection contains a ratio of (k-1)/k of dataset samples and the testing/validation collection contains a ratio of 1/k of dataset samples. The partitioning is done on the samples of each class separately. If stratified cross-validation is used, the number of samples for each class becomes equal in both training and testing datasets. Cross-validation techniques are generally named according to the number of partitions. For cross-validation and stratified cross-validation, the naming is given as k-fold cross validation and k-fold stratified cross-validation.

The dataset partitioning for 3-fold cross-validation is demonstrated in Figure 3.7**Error! Reference source not found.**. In this example, three authors with the numbers of 75, 45, and 150 documents are partitioned into three datasets, which are formed from training and testing sets in equal number of documents.

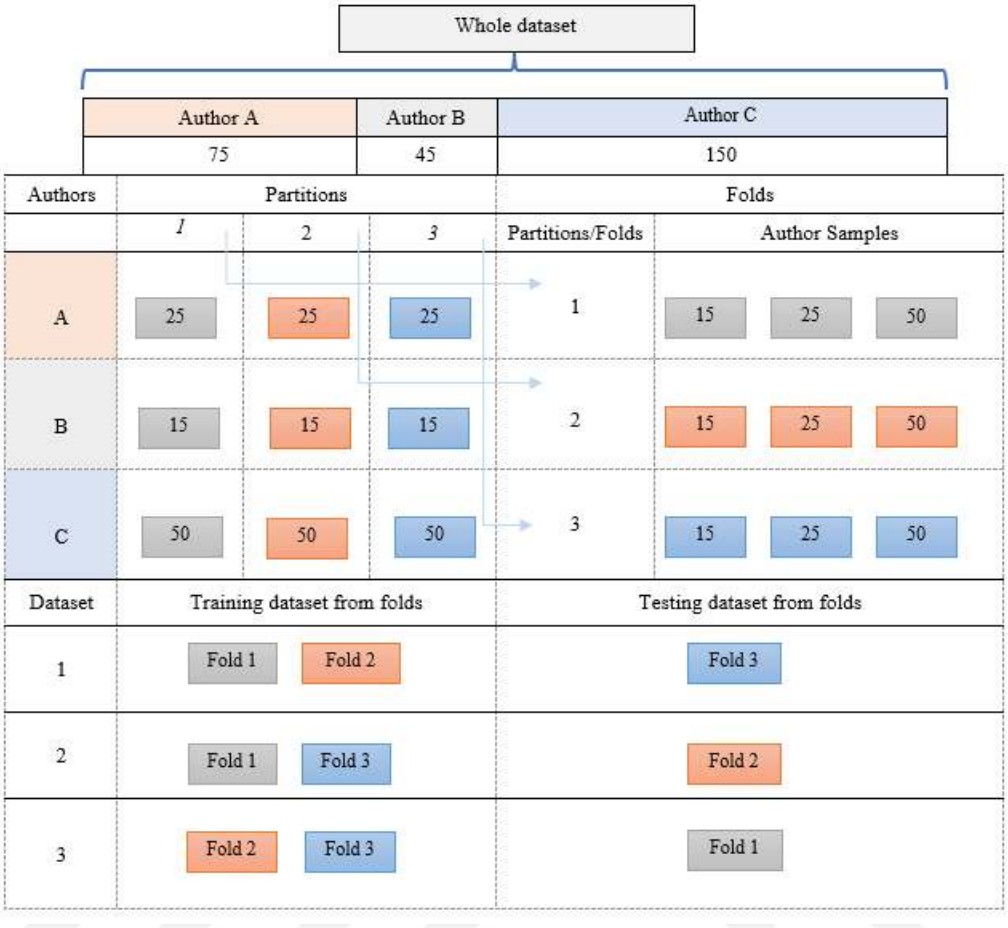


Figure 3.7. *Example for 3-fold cross-validation*

Figure 3.7 shows the number of samples for each author and for each partition. The construction of the final training and testing/validation datasets are also demonstrated according to the partition numbers. In this example, each author's samples are separated into 3 disjoint partitions randomly. Next, these partitions are combined, forming the fold groups. For each author, the disjoint sets are separated into 3 folds so that each fold contains the disjoint sets from all authors. The 2/3 ratio of the folds is used for training, and 1/3 ratio of the folds is used for testing. Since each fold contains the same ratio of samples for each author, they have the same number of documents. However, there are cases where the author's samples are not divisible by the number of folds, in which case the constructed datasets may not have equal number of samples.

In stratified cross-validation, the samples of each author are selected in equal numbers in order to reduce the class-imbalance problem during training and testing. For example, in an authorship attribution dataset, suppose that there are 1000 number of documents for the first author and 10 number of documents for the second author. The

high divergence rate of the number of training samples for each author confuses the optimization method during classifier training, and the classifier becomes balanced for the first author. On the other hand, in evaluation, the performance for the second author is negligible when compared to the first since the second author has very few numbers of testing samples compared to the first author. In order to prevent such problems, stratified cross-validation is preferred.

In cross-validation, evaluation is done on each training and testing dataset pairs separately. The general performance of a model is computed by averaging the evaluation scores for each label separately in all experiments. The final score of all experimental evaluations is computed based on macro or micro averaging. In macro averaging, each label is treated equally, and the final average is the mean of each label score. In micro averaging, however, the label scores are weighted based on the number documents in the labels, and the weighted mean is calculated. The details of evaluation measures and the averaging are explained in the following sections.

3.6. Evaluation Measures

In multiclass text classification, the performance evaluation requires different sets of measures in order to justify the accuracy of the proposed approach. Among such measures, accuracy, precision, recall and F-Measure are widely applied. Given a label, the prediction label and the instance label, there are four conditions appearing with a combination of positive or negative and true or false cases. These conditions are tested for each label separately. The final score of a condition is computed for each label by counting the number of test instances that satisfies the condition.

A true positive condition occurs when the given label, the prediction label and the original label of the instance are the same. A false positive condition occurs when the prediction label is equal to the given label; however, the original label of the instance is different. On the other hand, a true negative condition occurs when the given label is different from the prediction label and the instance label while the prediction label and the instance label are the same. In a false negative condition, the given label and the instance label are the same, but they are different from the prediction label. In Table 3.12, a demonstration of each condition for the given label “A”, and the ten instances with the labels A, B and C are given.

Table 3.12. *Example of Evaluation Measures for the given label "A"*

Prediction Labels	Instance Labels	True Positive	False Positive	True Negative	False Negative
A	A	✓			
A	B		✓		
B	B			✓	
C	C			✓	
C	A				✓
A	A	✓			
A	C		✓		
C	C			✓	
B	A				✓
A	A	✓			

In Table 3.12, the conditions true positive, false positive, true negative and false negative for the specified label "A" are checked according to the prediction and instance labels of ten samples. The rates of true positives, false positives, true negatives and false negatives are 0.3, 0.2, 0.3 and 0.2 respectively. The final prediction performance is computed for each evaluation measure separately. In the next subsection, the formulations for the evaluation measures accuracy, precision, recall and f-measure are given.

3.6.1. Final measures

The multiclass classification evaluation measures include accuracy, precision, recall and f-measure. Each of these measures combines the number of true positives (tp), false positives (fp), true negatives (tn) and false negatives (fn) differently. Accuracy and f-measure are widely applied in text classification. A common evaluation measure applied in multiclass classification tasks, accuracy measures the prediction performance according to both positive and negative cases. Therefore, it is robust in imbalanced datasets. The formulation of accuracy is given in Eq. (3.23).

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{3.23}$$

In Eq. (3.23), each condition is given by the number of instances that meet the condition. According to Table 3.12, the number of true positives, true negatives, false positives, and false negatives are 3, 2, 3 and 2 respectively, and the accuracy of label "A"

is 50%. Along with accuracy, F-Measure has been widely applied in multiclass text classification tasks. It is the harmonic mean of precision and recall as given in Eq. (3.24).

$$f - measure = \frac{2 * precision * recall}{precision + recall} \tag{3.24}$$

Precision is applied to measure the accuracy of the model in positive cases where the prediction is equal to the given label. On the other hand, recall evaluates the accuracy of a label based on measuring how many of the instances with the given label are predicted as correct. Therefore, precision and recall are completely different measures. The precision and recall formulations are given in Eq. (3.25) and Eq. (3.26) respectively.

$$precision = \frac{tp}{tp + fp} \tag{3.25}$$

$$recall = \frac{tp}{tp + fn} \tag{3.26}$$

According to Table 3.12, the precision and the recall of label "A" is computed as 50% and 60% respectively. Since the number of false positives of label "A" is higher than the number of false negatives, the precision is lower than the recall. Given the precision and recall scores as 50% and 60%, then F-Measure score of label "A" becomes 54.54%.

3.6.2. Averaging

Based on several experiments, the final score of the experimental evaluations is done through micro or macro averaging of the measures of each label. Micro averaging gives equal weights to each instance prediction whereas macro averaging gives equal weights to each label. Micro averaging is useful when the number of instances for each label is different, while macro averaging is useful when the evaluation for each label is important. In Eq. (3.27), micro averaging uses the number of instances for each label (N_L) for weighting each label score. In Eq. (3.28), macro averaging averages the score of each label equally. Both averaging methods may use either the number of conditions, such as true positive, false positive, true negative and false negative or the final scores, such as accuracy, f-measure, precision, and recall in computation of the averages.

$$micro = \frac{\sum_L (N_L * score_L)}{\sum_L N_L} \tag{3.27}$$

$$macro = \frac{\sum_L score_L}{number\ of\ labels\ (L)} \tag{3.28}$$

3.7. Bootstrap sampling

The common approach before applying a significance test is to increment the number of observations for the experimental evaluations. Since experimental observations is costly, sampling techniques are applied to virtually increment the observations. However, the significance test is only valid when there is enough evaluation scores. In order to increment the number of performance samples of a given experimental setup, sampling is applied. There are two sampling scenarios where either the number of test samples is sampled or the number of evaluation scores is sampled. In this thesis, the second scenario is applied to increase the number of evaluation scores. A general overview of the sampling technique is given in Figure 3.8.

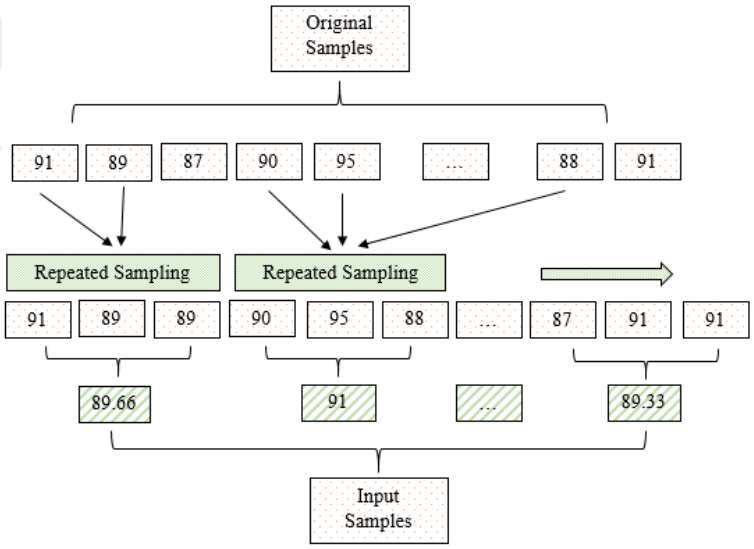


Figure 3.8. *Bootstrap Sampling*

Figure 3.8 shows the steps of bootstrap sampling from top to bottom. In bootstrap sampling, two parameters are applied to randomly select the samples, which are the number of samples in each selection stage and the number of selection stages. In common application of bootstrap sampling, the samples are selected with repetition for a fixed number of times and are combined by computing their means. This selection process is repeated several times. After sampling, the number of final samples becomes equal to the

number of selection stages. In Figure 3.8, original samples are selected for three times with repetition. The selection stage is also repeated many times. In each selection, the samples are averaged, and they become the new input samples.

3.8. Significance test

The hypothesis tests, such as student's t-test and Mann–Whitney U test are common tests for measuring the significance of the applied method. These significance tests measure the likelihood of the divergence between two different distributions. Significance occurs when the null hypothesis is rejected. In student's t-tests, the null hypothesis states that the mean of the divergence is zero. In Wilkinson's test, the null hypothesis states that the random samples drawn from two distributions should be equal or equally likely. In the significance testing, the final evaluation measures are used either directly or via bootstrap sampling. In our evaluations, we applied student's t-test on the paired mean F-Measure scores obtained by bootstrap sampling.

4. COMMON VECTOR SCALING

Due to varying text lengths, topics, and genres, the heterogeneous nature of web documents have negative effects on the performance of AA. Proposed solutions that combine a diverse set of features into a single document vector do not overcome this problem. Most stylometric features have domain dependencies; for instance, features used in the prediction of genres are also available for author prediction. In this thesis, a scaling approach for removing domain-specific properties, which are similar to background subtraction tasks in image recognition, are proposed. Our approach consists of two steps: determination of domain vectors and calculation of the differences between document and domain vectors. In the first step, the domain vector is determined. The common vectors for each domain via the CVA are simple computed. In the second step, the domain vector and the document vector are bucketized. Then, the bucketed domain vector is subtracted from the bucketed document vector. The final vector is a difference vector, and it represents the classifier input of the specified document. In the classification model, this vector for both training and testing are used. Since scaling is a vector-subtraction process, it does not create inefficiency during parameter optimization in classifier training or during prediction in testing. To the best of our knowledge, this is the first scaling technique that is applicable to any text classification problem adversely affected by the domain.

4.1. Methodology

Domain invariant properties are extracted from the documents of the domain using the CVA. Scaling is performed through bucketed subtraction of the domain vector from the document vector. Bucketization is used to cancel out similar values during subtraction. In Figure 4.1 and Figure 4.2, the process of bucketed common vector scaling is illustrated.

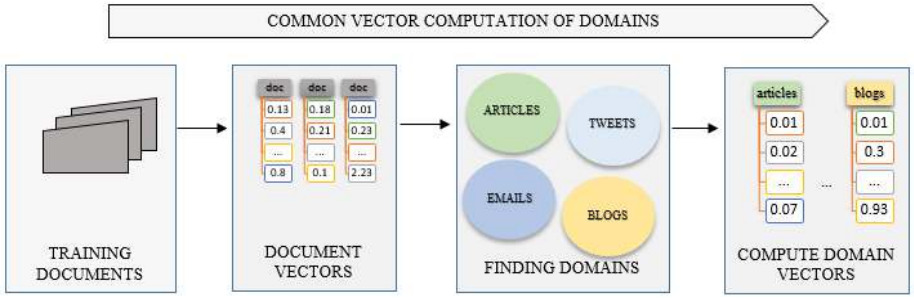


Figure 4.1. *Schematic diagram of the bucketed common vector computation*

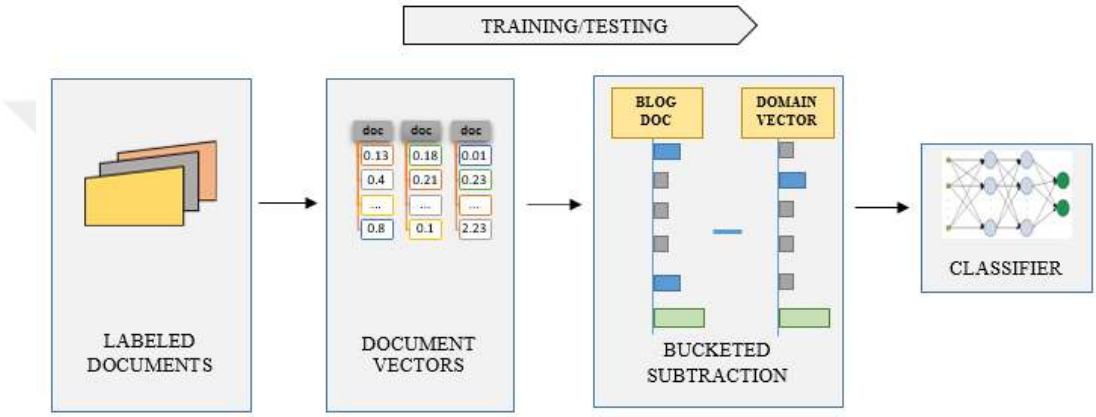


Figure 4.2. *Schematic diagram of training with scaling*

In Figure 4.1, the first step consists of grouping all the documents according to the domain. The common vectors for each domain are computed separately by the CVA. In the second step, given in Figure 4.2, each document is scaled by bucketed subtraction, where the difference between the document and its domain vector is computed. Finally, difference vectors are used as input in both training and testing of the classification method.

4.2. Computing the Common Vector

The CVA is a technique applied in speech and image recognition tasks to model class-invariant properties. It is used for background subtraction [196], face recognition [197] and isolated speech recognition tasks [44]. As discussed in the introduction, the CVA is an efficient strategy for differentiating unique author properties from the domain. The common vector for each domain is computed using the Gram-Schmidt orthogonalization process. Each document in a specified domain is represented by the

TF-IDF document vector. Then, a column matrix is obtained from each document vector. Using the Gram-Schmidt process, the orthonormal Q matrix is used. The Q matrix is used to project a sample of the domain. The common vector is obtained by subtracting the projection vector from that sample. For convenience, the Scala breeze library [198] is used to compute the Q matrix. An example of a common vector computation for two documents is presented in Eq. (4.1) to Eq. (4.5).

Let $d1 = [5.0, 2.0, 3.0]^T$ and $d2 = [1.0, 2.0, 0.0]^T$ are two document vectors from the same domain. The column matrix of these two document vectors is constructed in Eq. (4.1).

$$M = \begin{bmatrix} 5.0 & 1.0 \\ 2.0 & 2.0 \\ 3.0 & 0.0 \end{bmatrix} \tag{4.1}$$

First, the difference vectors are computed by subtracting the first column vector from the other columns of matrix M. Then, matrix Q is obtained using the Gram-Schmidt orthogonalization process.

$$b_1 = d_2 - d_1 = [-4.0, 0.0, -3.0]^T \tag{4.2}$$

In Eq. (4.2), b_1 is called the difference vector. In Eq. (4.3), z_1 is the unit vector in the direction of b_1.

$$z_1 = \frac{b_1}{||b_1||} = [-0.8, 0.0, -0.6]^T \tag{4.3}$$

The columns of Q are represented by orthogonal vectors $(z_1 \ldots z_n)$, which are obtained via orthogonalization, as in Eq. (4.3). Matrix Q is given in Eq. (4.4).

$$Q = \begin{bmatrix} -0.8 \\ 0.0 \\ -0.6 \end{bmatrix} \tag{4.4}$$

The common vector is obtained by subtracting the projection of any document vector of the given domain from the orthonormal subspace spanned by the columns of Q, as stated in Eq. (4.5).

$$\begin{aligned} cv &= d_1 - Q(Q^T Q)^{-1} Q^T d_1 \\ &= d_1 - \langle d_1, z_1 \rangle z_1 \\ &= [5.0,2.0,3.0] - [2.64, 0.0, 1.98] \\ &= [2.36, 2.0, 1.02] \end{aligned} \tag{4.5}$$

In this example, the common values of the two document vectors (the second index) are preserved in the common vector.

4.3. Bucketization and Subtraction

Bucketization and subtraction are the final steps before training and testing. For bucketization and subtraction, the scaled differences between the domain vector and the document vector are measured. Bucketization places important terms and unimportant terms into corresponding bins in such a way that subtraction cancels them out. Vector bucketization is illustrated in Figure 4.3.

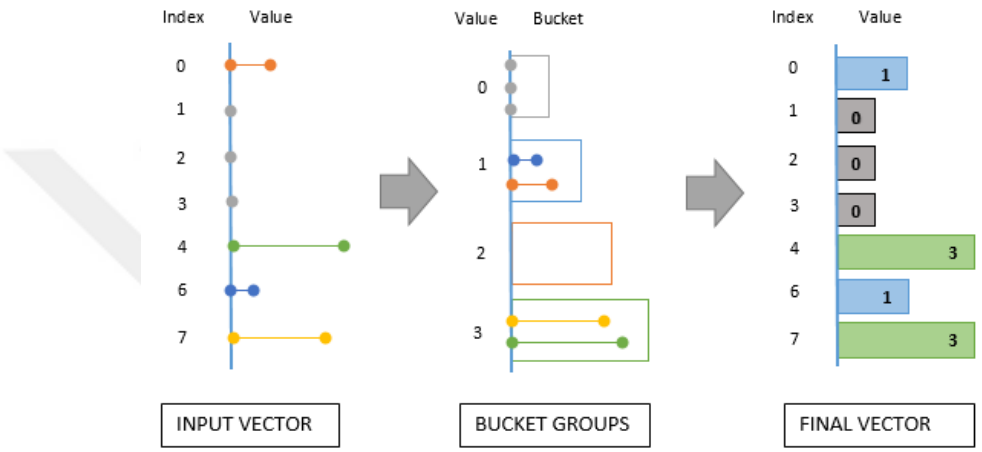


Figure 4.3. *Bucketized document vector*

In Figure 4.3, first the vector values are grouped into buckets, which is known as grouping. In the second stage, each index is assigned the values of the corresponding group to which the vector value belongs. Bucketization facilitates transformation of the document from a TF-IDF vector to a scaled categorized vector. After subtraction, the final vector is represented by the differences of each feature from the domain. The values in the differences are proportional to their occurrence rates in the document.

4.4. Implementation Framework

In this section, the details of the experimental framework are explained. In text classification, each document is processed by a pipeline model, which is a directed acyclic graph where a separate text processing technique is attached to its predecessor and executed in the sequential order. For instance, tokenization is applied to the document before extracting token suffixes, and token suffixes are applied before frequency extraction of each suffix. Every text-processing pipeline contains a global aggregation

step to compute the models and the outputs. The general processing pipeline of a text classification system contains aggregation steps of dictionary construction and classification model optimization. In this pipeline, various numbers of steps may use different kinds of language processing tools and provide a diverse set of outputs.

Traditional text classification approaches are designed with in-memory computational models, where each processing step is referenced in memory during the whole computation of a general pipeline. Such approaches suffer from lack of memory when large outputs are formed by any of the pipeline step. For this reason, an in-memory map-reduce framework is applied to extract features and training the classification algorithm. In this framework, the data is in row and column format, where each cell is attached to a specified processing pipeline. In any step of the pipeline, either previously computed data or re-computed from a cached version is used so that only necessary blocks are referenced in memory. This execution model is known as lazy evaluation. Lazy evaluation applies the prior processing steps when only referenced data is needed for an output.

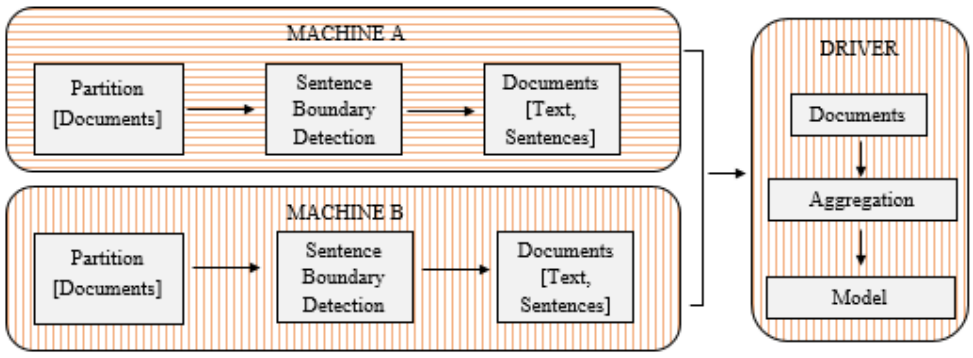


Figure 4.4. *A simple pipeline with aggregation*

A simple pipeline orientation is shown in Figure 4.4. This orientation is a simple overview of a distributed in-memory map-reduce computation model for a text classification system. In this orientation, documents are partitioned based on locality, in which a partition is locally loaded to the machine. The sentence boundary detection step is distributed across the machines and executed in parallel. The final output of this step contains documents with sentences. After the execution is finished, a driver program gathers the results from each machine and builds a model. This model may contain the

statistics about the sentences in all the documents. For example, the average sentence length is an example of such statistics in readability measures.

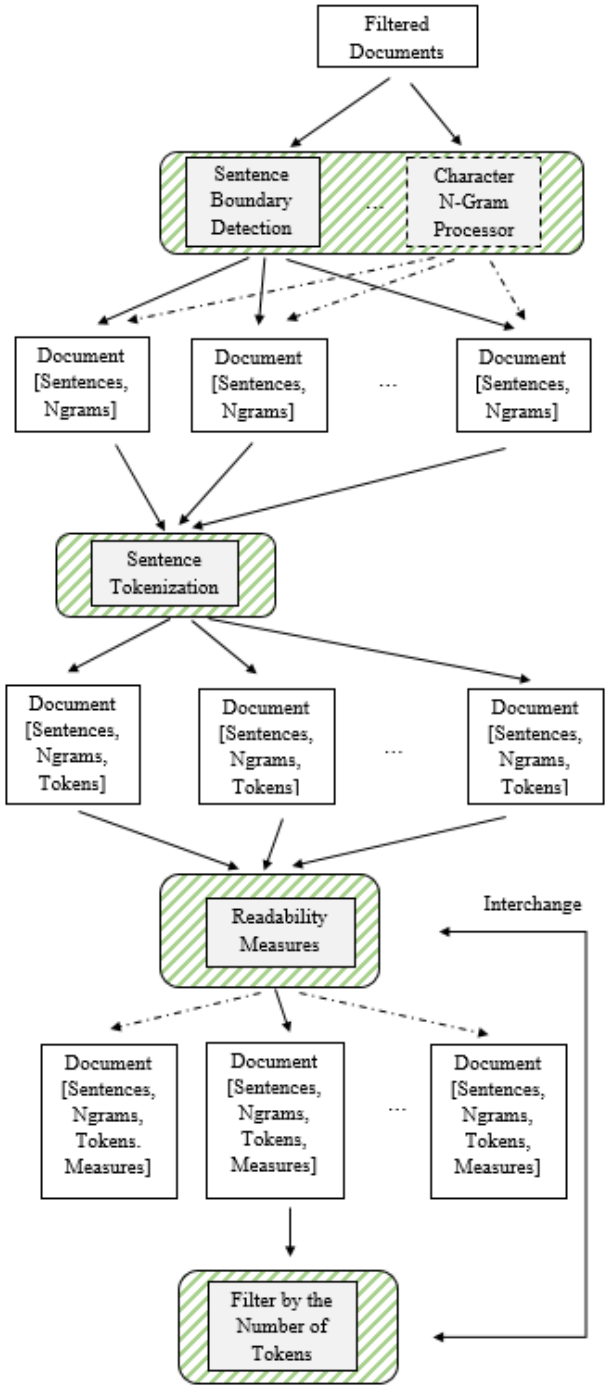


Figure 4.5. *Lazy Evaluation Example*

In Figure 4.5, a pipeline model with three processing steps and one filtering step is shown. This example processes the filtered documents and extracts the documents in a

row representation with columns of sentences, ngrams, tokens, and measures. The extracted documents are filtered based on the number of tokens. In the figure, each processing step is shown with dashed background. These steps are operated as in Figure 4.4. However, according the lazy evaluation principal, only referenced cells from the final output are processed. According to this pipeline, the documents with sentences, tokens and measures are the final outputs. Therefore, the referenced cells are from the documents with a certain number of tokens and from the columns of sentences, tokens and measures. Since the n-grams do not have any reference in the output, its processing step is not applied. Moreover, the physical execution can be translated into a more efficient logical execution by interchanging the processing steps of filtering and readability measures so that not all documents are processed by the readability measures step. In order to show the unused connections in the logical execution given in Figure 4.5, dashed arrows were used.

4.5. Implementation Details

All the experimental evaluations are modelled based on the text-processing pipeline given in three stages: document preparation, dataset partitioning, document vector extraction and evaluation. The document preparation stage includes the steps of loading, cleaning and filtering the documents. Partitioning stage constructs the training and testing datasets according to the evaluation strategy. Feature extraction stage contains feature extraction, feature weighting, feature scaling, and classifier training. The evaluation stage contains the evaluation of the classification model.

In Figure 4.6, the document preparation stage is demonstrated. The document preparations start with loading the document, cleaning the data and masking the author relevant information. In these steps, the text data is first converted into a structured document object. Each document object contains the fields of text, genre, domain and author name. The data cleaning step removes the documents with missing or inconsistent field values. The twitter documents with replies, retweets and urls are eliminated in this step. Also, the blog and article documents with the text field that contains only number, urls or html tags are removed from the collection. The documents of diverse resources may contain a different naming for the same set of genres. In order to eliminate this inconsistency, the genre names are mapped by a lookup table. After data cleaning is done, emails, urls, phone numbers and addresses in the text filed of the remaining documents

are masked. A common masking technique is to remove the paragraphs which contain n-grams of the author name, email addresses and/or the link.

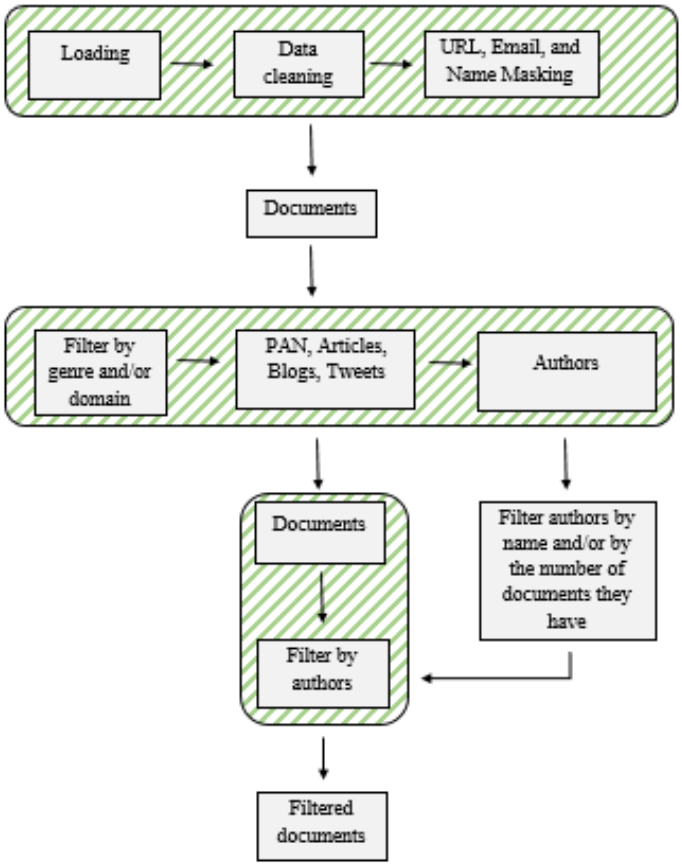


Figure 4.6. *Document preparation stage*

In the second step of the document preparation stage (Figure 4.6), the documents are selected by using predefined set of genres, authors and/or domains. This step includes three sub-steps where the genre/domain filtering, author selection and document filtering occur in order. Firstly, the documents are filtered according to genre and/or domain. Secondly, the authors in the given author set and/or the authors with at least a fixed number of documents are selected. Thirdly, the documents of these authors are filtered.

In Figure 4.7, the dataset partitioning stage is demonstrated. In this stage, the output of the document preparation stage is used in construction of the datasets. Firstly, the documents with the character length in a predefined range are filtered. Secondly, the top authors of these documents are extracted by selecting the authors with the highest number of documents. The authors in the top list are used to filter the documents, and finally these documents are used in the dataset partitioning. Dataset partitioning involves the steps of

k-fold cross validation, in which the documents are partitioned into k disjoint sets. The partitioning is done according to the steps given in Section 3.6.

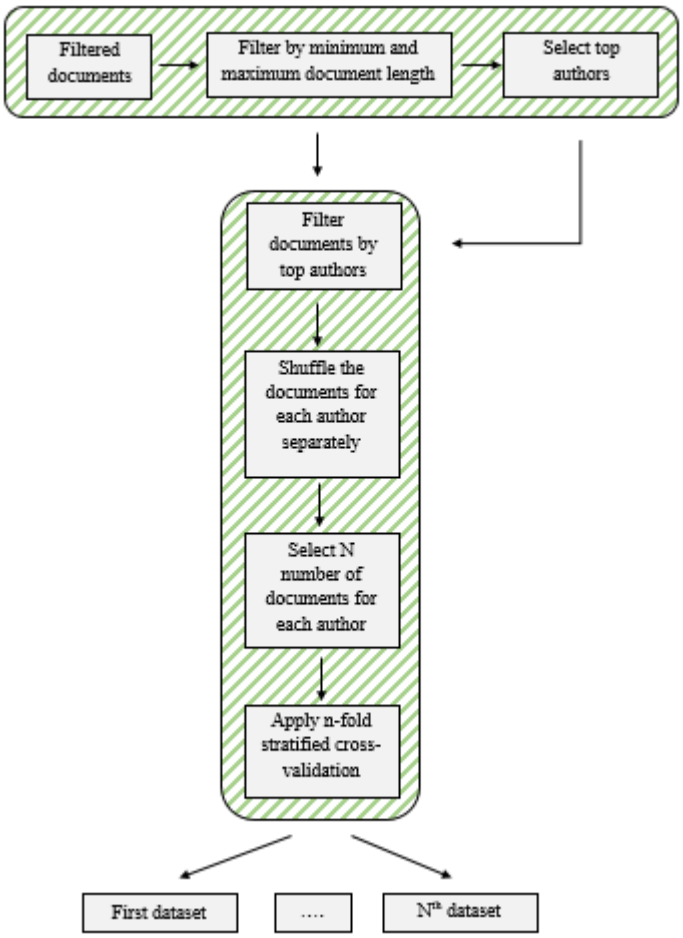


Figure 4.7. *Dataset partitioning stage*

In Figure 4.7, the datasets created in dataset partitioning stage contain both training and testing documents. The training documents of these datasets are used in the feature extraction stage as given in Figure 4.8. The feature extraction stage extracts the features separately based on the corresponding feature extraction process. Each feature extraction process may have different sub-steps, such as weighting and normalization. Each extraction step creates the necessary vocabulary and statistics for the feature vector extraction. These statistics are also used jointly by the common vector computation in the whole document collection. Common vectors are obtained for each genre and/or domain. The common vector computation is explained in Section 4.2. The common vectors are used to scale the final document vectors as explained in Section 4.3. After the document

vectors are obtained, the classification algorithm is trained. The classification algorithm extracts the necessary model parameters for the evaluation stage.

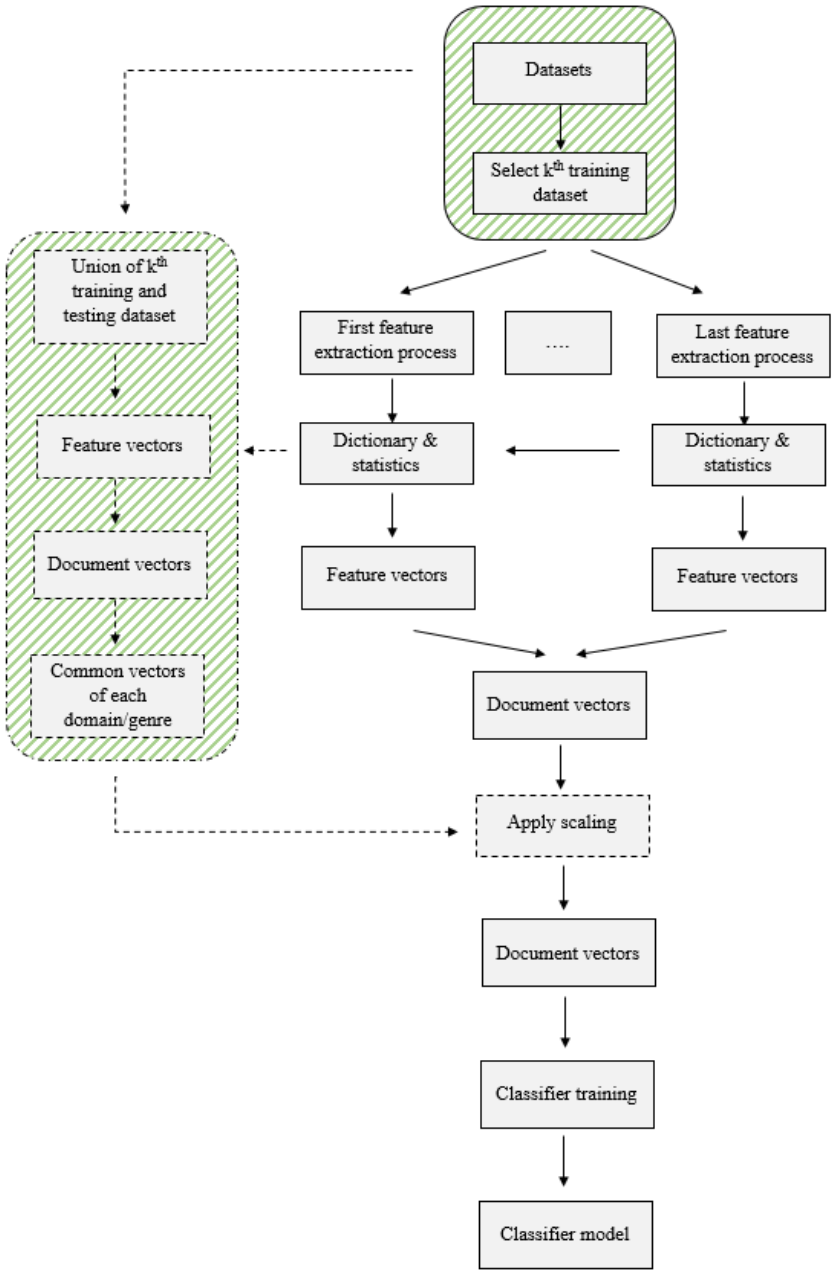


Figure 4.8. *Feature extraction stage*

In Figure 4.8, the common vector computation and scaling are optional, which is why they are represented by dashed lines. The datasets, dictionary & statistics, and common vectors created in the feature extraction stage are used in the evaluation stage

given in Figure 4.9. The evaluation stage is repeated for all the dataset separately, and the final evaluations are averaged according to the micro-averaging. Figure 4.9 demonstrates the evaluation stage for the k^{th} dataset and the corresponding inputs of dictionary & statistics, common vectors and classifier model.

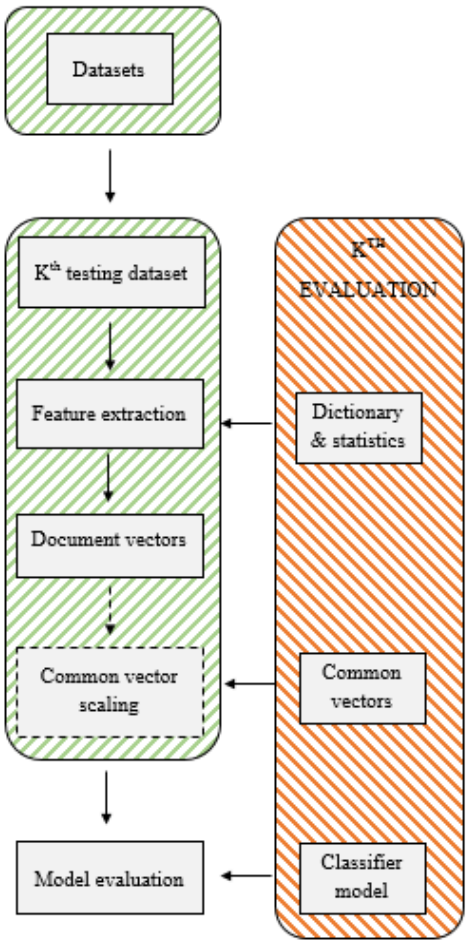


Figure 4.9. *Evaluation stage*

In Figure 4.9, the feature extraction steps use the dictionary & statistics in extraction of the document vectors for k^{th} test datasets. The feature extraction applies the steps of the feature extraction stage except for creating dictionary & statistics. After the document vectors are extracted, the corresponding common vectors are optionally used for scaling. Finally, the classifier model is evaluated for these document vectors.

5. EXPERIMENTAL SETUP

In this section, the details of the experimental evaluations are given. First, the parameter types and their assignments for each algorithm and for each dataset are introduced. For these parameters, two different performance analyses are constructed. The goal of the first analysis is to measure the robustness of different feature sets for increased number of authors. The second analysis is to measure the performance of author prediction in heterogeneous datasets. The results of the CV scaling for the classifiers, feature sets and datasets are also given in these analyses. In the following subsections, the parameter settings, the first and the second analyses are presented.

5.1. Parameters

In this section, the parameters for classifiers and CV scaling are given for the corresponding datasets in all the experimental setups. The classifier parameters are tuned based on the number of features, dataset size and the domains in the dataset. The bucket range – an important parameter applied in CV scaling – is adjusted based on the maximum and minimum values of the feature vector for each feature set and dataset. The CV scaling is applied to the best performing parameters without any specific adjustments on the bucket range.

Table 5.1. *Bucket range parameters*

Parameter name	Target datasets	Value
Bucket range	Short-length A10, Short-length A15, C10	[0, 10, 40]
Bucket range	Short-length A25, Short-length A40	[0, 20, 40]
Bucket range	Medium-length A10	[0, 5, 20]
Bucket range	Medium-length A15	[0, 5, 100]
Bucket range	Medium-length A25	[0, 2, 80]
Bucket range	Medium-length A40	[0, 2, 80]
Bucket range	ART-5, ART-10	[0, 20, 80]
Bucket range	BLOG-5	[0, 5, 50]
Bucket range	BLOG-10	[0, 20, 80]
Bucket range	TWE-5, TWE-10, PAN-10	[0, 5, 40]
Bucket range	ART+BLOGS, ART+TWE, PAN18-1	[0, 20, 40]
Bucket range	BLOG+TWE, ART+BLOG, PAN-5	[0, 20, 40]
Bucket range	C50	[0, 10, 100]
Bucket range	Others	[0, 5, 40]

In Table 5.1., the bucket parameters are specified for each dataset. The values of the bucket parameters are defined as a range, in which the first value is the minimum, the

second value is the maximum, and the last value is the number of buckets. For example, in a range defined as [0, 20, 40], the minimum value is 0, the maximum value is 20, and the number of buckets is 40. According to the bucketization, a value that is not in the range will be assigned to the closest value that is either the minimum or the maximum.

The classification parameters for the multinomial naïve Bayes, logistic regression (LOG), decision tree (DT), random forest (RF) and multilayer perceptron (MLP) classifiers and the dimensionality reduction rate for PCA algorithm are given in Table 5.2. The unspecified parameters are left as default values in the Spark 2.3.2 framework.

Table 5.2. *General parameters*

Algorithm	Dataset	Parameter Setting	Description
DT and RF classifiers	All datasets	Maximum BinSize = 80, Maximum Depth=16	Maximum number of bins and depth in decision tree learning
PCA	Competition datasets: PAN11, PAN18, C10, and C50	Number of dimensions = 100	Dimensionality reduction size (k)
LOG	All datasets	Standard Scaling	True
MLP	Datasets with dictionary size < 10000	Layers = (Input Size, 50, Label Size)	Layers and their sizes of MLP classifier
MLP	Datasets with dictionary size > 10000	Layers = (Input Size, 500, Label Size)	Layers and their sizes of MLP classifier
MLP	All datasets	Activations = (Sigmoid, Softmax)	Activation functions for hidden and output layers in MLP classifier
MLP	All datasets	Loss Function = CrossEntropy	Loss function computed in the output layer of MLP classifier
MLP, LOG	All datasets	Stopping tolerance= 1E-6	The minimum acceptable change in gradient during optimization

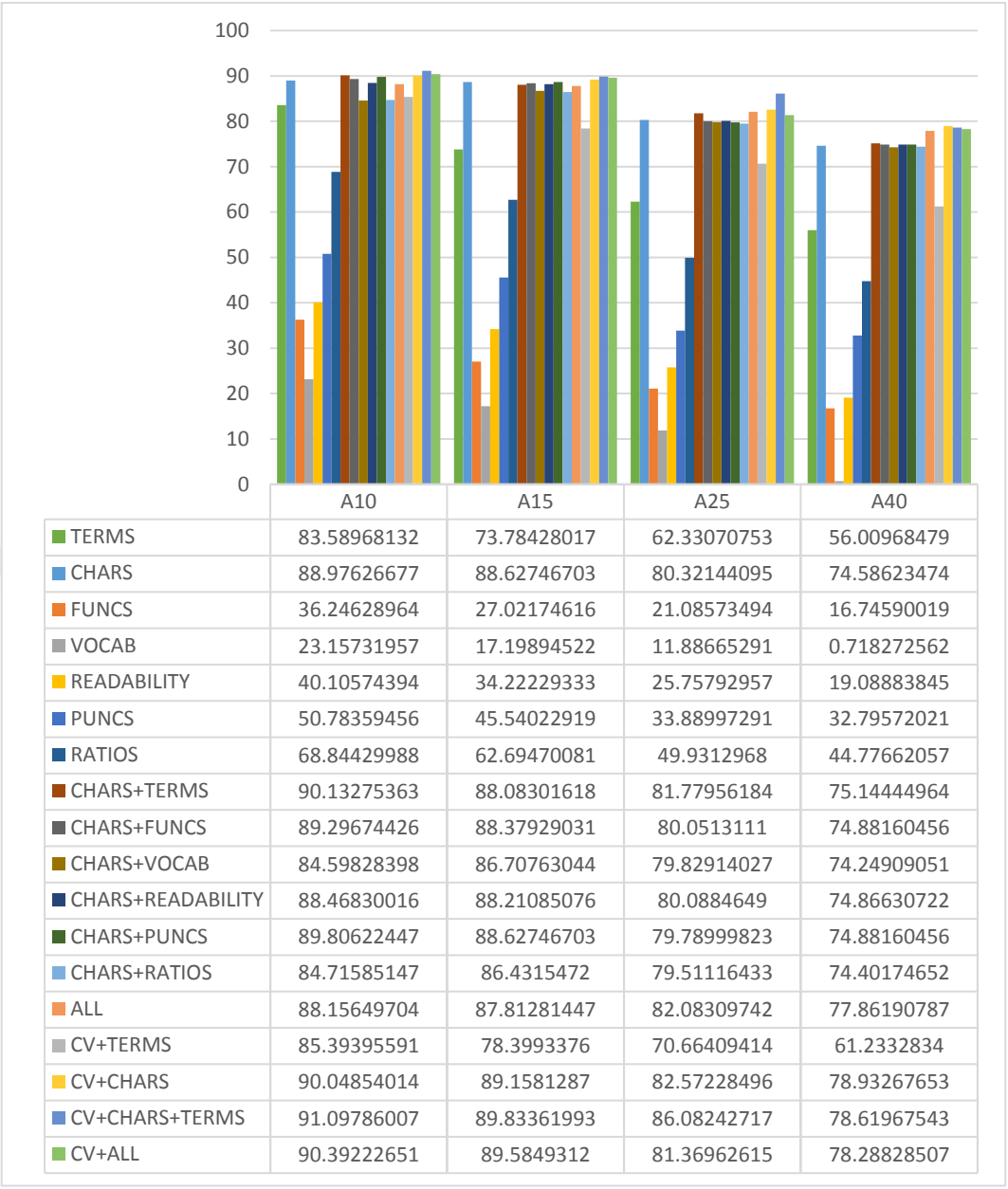
In Table 5.2, the algorithms applied in experimental evaluations are represented in the first column. In the second column, the corresponding datasets are indicated by their abbreviations or by the condition on the dictionary size. The dictionary size is important to determine the input layer size of the MLP classifier. For the datasets with the number of dictionary items greater than 10000, the hidden layer size for MLP classifier is set to 500, and for other datasets, it is set to 50. The output layer size of MLP classifier is determined by the number of authors in the dataset, and one hidden layer and one output layer are used in all network configurations. In addition, in each of these layers, sigmoid

and softmax activation functions are selected respectively. The optimization algorithm for LOG and MLP classifiers is limited-memory BFGS (L-BFGS). It is the default optimization routine in Spark 2.3.1 framework for these classifiers. For the PCA algorithm, the best performing dimensionality reduction size for all datasets is selected as 100 from the candidates of 500, 250 and 50.

5.2. The Effects of Size

In this section, the logistic regression performance of the common stylometric features in 3-fold cross validation strategy is compared. In the comparisons, two document collections sampled from blogs according to different document lengths are used. The documents of the first collection have shorter lengths than the documents in the second collection. The first collection is sampled from the documents with a minimum character length of 200 and a maximum character length of 1000. The second collection is sampled from the documents with a minimum character length of 2000 and maximum character length of 7000. The first collection consists of short-length documents, and the second collection consists of medium-length documents. Both of these collections are compiled from Turkish blog documents. In Figure 5.1 and Figure 5.2, the short-length and the medium-length document collections are used respectively to compare the effects of the number of authors.

Four datasets from both of these document collections are compiled with 10, 15, 25 and 40 number of authors. All the datasets contain 40 training and 20 testing documents for each author. The results given in Figure 5.1 and Figure 5.2 consist of the f-measure scores of the logistic regression classifier for 3-fold stratified cross validation. In these figures, the results for stylometric features of character n-grams (chars), terms (terms), ratios (length ratios), punctuations (puncs), readability measures (readability), vocabulary richness measures (vocab) and function words (funcs) are given. The combination of all the single feature sets is given by “all” and bi-combinations of these features is given by plus (+) symbol. Only the best performing single feature set is combined with other features. In addition to the feature set comparisons, the CV scaling is applied for chars, terms and all feature sets. The results of the CV scaling applied on a feature set are given by plus (+) symbol.

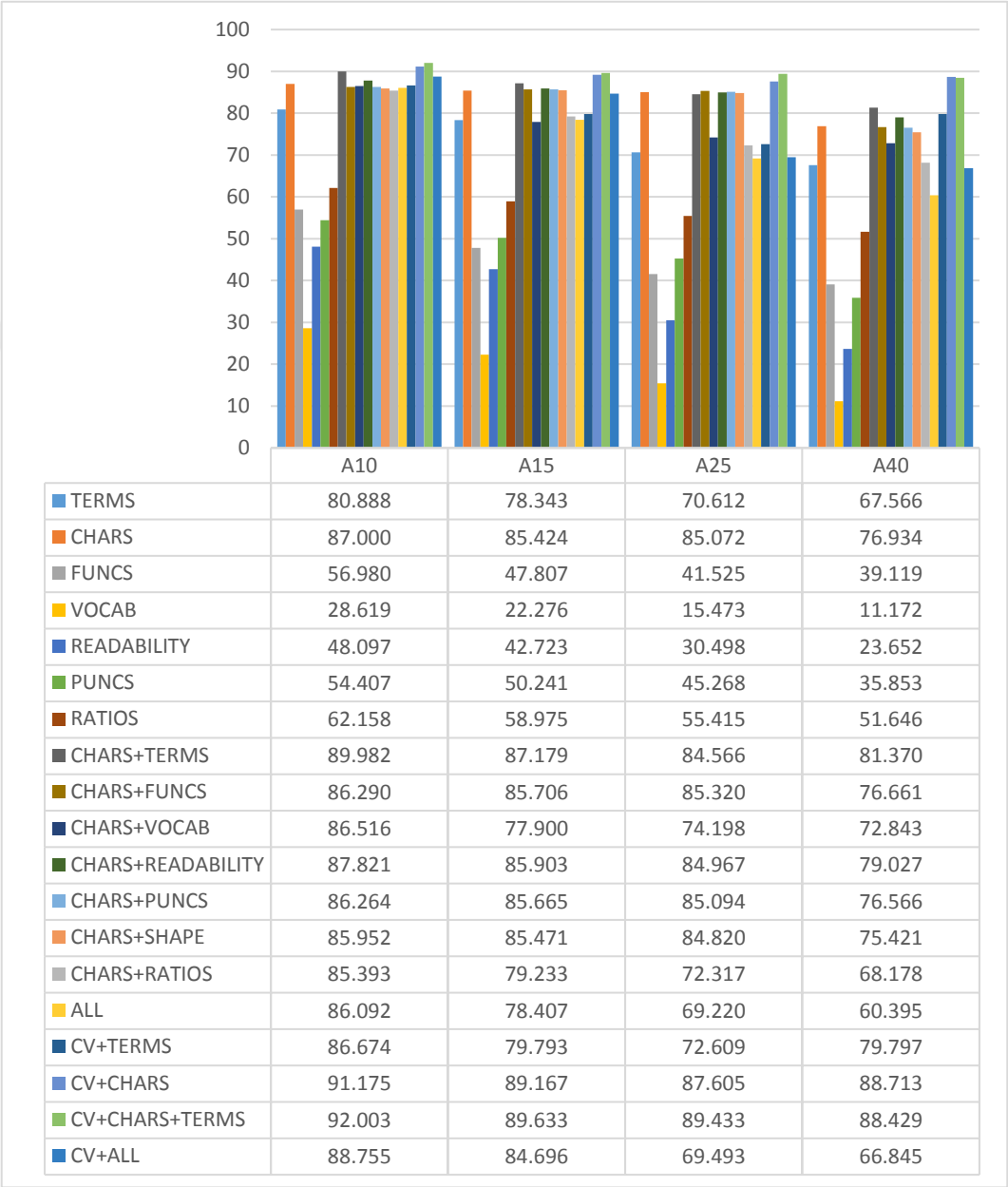


	A10	A15	A25	A40
TERMS	83.58968132	73.78428017	62.33070753	56.00968479
CHARS	88.97626677	88.62746703	80.32144095	74.58623474
FUNCS	36.24628964	27.02174616	21.08573494	16.74590019
VOCAB	23.15731957	17.19894522	11.88665291	0.718272562
READABILITY	40.10574394	34.22229333	25.75792957	19.08883845
PUNCS	50.78359456	45.54022919	33.88997291	32.79572021
RATIOS	68.84429988	62.69470081	49.9312968	44.77662057
CHARS+TERMS	90.13275363	88.08301618	81.77956184	75.14444964
CHARS+FUNCS	89.29674426	88.37929031	80.0513111	74.88160456
CHARS+VOCAB	84.59828398	86.70763044	79.82914027	74.24909051
CHARS+READABILITY	88.46830016	88.21085076	80.0884649	74.86630722
CHARS+PUNCS	89.80622447	88.62746703	79.78999823	74.88160456
CHARS+RATIOS	84.71585147	86.4315472	79.51116433	74.40174652
ALL	88.15649704	87.81281447	82.08309742	77.86190787
CV+TERMS	85.39395591	78.3993376	70.66409414	61.2332834
CV+CHARS	90.04854014	89.1581287	82.57228496	78.93267653
CV+CHARS+TERMS	91.09786007	89.83361993	86.08242717	78.61967543
CV+ALL	90.39222651	89.5849312	81.36962615	78.28828507

Figure 5.1. *Author size effect for the short-length documents*

In Figure 5.1, the results are given for the datasets compiled from short-length documents. In all the datasets, the best performance is obtained by the CV scaling. The CV scaling is not as effective on the combination of all the features as on the chars or term features. According to these results, the second best performance is obtained by the combination of the all features, and the worst performance is obtained by vocab feature set. In all the datasets, the rank of the top performing single features is chars, terms, ratios, puncs, readability, funcs and vocab. On the other hand, the performance of the combined features is significantly affected by the number of authors. For instance, in A10 and A15

datasets, the second best performance is obtained by chars + terms and chars + puncs feature set respectively, while in A25 and A40, the second best performance is obtained by the combination of all features.



	A10	A15	A25	A40
TERMS	80.888	78.343	70.612	67.566
CHARS	87.000	85.424	85.072	76.934
FUNCS	56.980	47.807	41.525	39.119
VOCAB	28.619	22.276	15.473	11.172
READABILITY	48.097	42.723	30.498	23.652
PUNCS	54.407	50.241	45.268	35.853
RATIOS	62.158	58.975	55.415	51.646
CHARS+TERMS	89.982	87.179	84.566	81.370
CHARS+FUNCS	86.290	85.706	85.320	76.661
CHARS+VOCAB	86.516	77.900	74.198	72.843
CHARS+READABILITY	87.821	85.903	84.967	79.027
CHARS+PUNCS	86.264	85.665	85.094	76.566
CHARS+SHAPE	85.952	85.471	84.820	75.421
CHARS+RATIOS	85.393	79.233	72.317	68.178
ALL	86.092	78.407	69.220	60.395
CV+TERMS	86.674	79.793	72.609	79.797
CV+CHARS	91.175	89.167	87.605	88.713
CV+CHARS+TERMS	92.003	89.633	89.433	88.429
CV+ALL	88.755	84.696	69.493	66.845

Figure 5.2. *Author size effect for the medium-length documents*

In Figure 5.2, the results are given for the datasets compiled from medium-length documents. In all the datasets, the worst performance is obtained by vocab feature set. The rank of the top performing single features is affected by the author size. The highest performance degradation is observed in readability feature set, and the lowest

performance degradation is observed in chars + terms feature set. The overall performance of the chars + terms feature set is significantly higher than the combination of all features. The CV scaling improved the performance of terms, chars and all combination of features, and it achieved the best performance in all the datasets.

Based on the results given in Figure 5.1 and Figure 5.2, although the top performing ranks are different, the scores are similar for the A10 and A15 datasets. For A25 and A40 datasets, the scores for the medium-length documents are higher than short-length documents, which shows that the increased length of the documents becomes effective on the performance when the author size increases. The most performance increase is observed in chars + terms feature set by around 6 points. All feature combinations, chars + vocab and chars + ratios features are negatively affected by the document length. Moreover, in medium-length datasets, the performance improvement of the CV scaling is higher than the short-length datasets. For instance, the performance improvement of CV scaling on term features for the A40 dataset is around 12 points in medium-length documents, while it is around 5 points for short-length documents.

5.3. The Effects of Domains

In this section, we analyze the performance of CV scaling on multi-topic datasets compiled from Turkish and English sources. In the subsections of 5.3.1 and 5.3.2, the details and the statistics of these datasets are given respectively. In Section 5.3.3 and 5.3.4, the classification performances of the CV scaling approach on term and character n-grams features are compared.

5.3.1. Datasets

Our datasets are grouped into two main categories based on evaluation strategy. The first group of datasets consists of the training and testing samples of the PAN datasets: C10, C50, PAN11 and PAN18 [199, 200]. C10 and C50 datasets contain Reuter's newswire stories which cover corporate and industrial topics. The PAN11 dataset contains two sets of collections, namely large and small. Both collections are sampled from a subset of an email collection known as ENRON corpus [148]. PAN18 contains two sets of document collections sampled from novels. Both collections have training and evaluation datasets in distinct domains. PAN18 is the first cross-domain AA dataset. The second group of datasets is compiled by sampling Turkish web documents

and the large collection in PAN11 dataset. Turkish web documents have been crawled from blogs, tweets and journal articles for approximately two years at specified intervals. This prevents the topical imbalance caused by significant events in history, such as the failed Turkish coup on July 15 and the U.S. presidential election. In the processing stage of the web documents, relevant author information is masked via regular expression and named-entity-tagger-based filtering on urls, emails, addresses and names. Tweets with replies, urls and retweets are eliminated during this process. For PAN datasets, the author-related names, urls, emails and addresses are already masked, and thus no processing is performed.

For both the large PAN11 and Turkish datasets, the sampling strategies for the training and testing datasets are the same. First, via sampling, a specified number of documents from the top authors are randomly selected. The authors are sorted in reverse order of the associated number of documents in the collection. The authors of the large PAN dataset are selected from the top authors in the union of the training and testing collections. For mixed-source Turkish datasets, the authors are filtered manually, and top authors are selected from the intersection of two sources, such as blogs and tweets or articles and tweets. Second, a stratified 10-fold cross-validation technique is used with 120 documents per author to balance the number of documents for each author for both training and testing datasets. Hence, 90% of the documents of each author are randomly selected for training and 10% for testing. The general distributions of these sampled datasets are given in the next subsection.

5.3.2. Datasets by numbers

Different characteristics of each dataset affect the classification performance of authors. A shallow analysis of the topic, source and author distributions of each dataset are given in Table 5.3. The general properties of these datasets are explained in Section 5.3.1. The datasets compiled from PAN, articles, blogs and tweets are abbreviated by PAN, ART, BLOG, and TWE respectively. The number of authors is separated by the dash (-) symbol, and the source combinations are given by the plus (+) symbol for each of these datasets The competition datasets are abbreviated according to the name of the competition, data and the problem type. For instance, PAN18-1 is the abbreviation of the PAN author identification competition at 2018 for the first task. The statistics for sampled datasets with specified numbers of authors are obtained for the union of 10 test datasets

generated by 10-fold stratified cross-validation. The statistics for competition datasets are obtained from the union of training and testing partitions.

Table 5.3. *Dataset statistics*

Dataset	Author count	Document count	Token count	Avg. authors per topic	Avg. topics per author	Avg. authors per source	Avg. sources per author
PAN11-L Train	72	10635	32069	35	2.430	-	-
PAN11-L Test	72	1300	9424	16.8	1.412	-	-
PAN11-S Train	26	3519	14936	9.6	1.846	-	-
PAN11-S Test	26	495	4778	12.5	1.186	-	-
PAN18-1 Train	20	140	10480	-	-	-	-
PAN18-1 Test	20	105	8235	-	-	-	-
PAN18-2 Train	5	35	4662	-	-	-	-
PAN18-2 Test	5	21	3255	-	-	-	-
PAN18 1&2 Train	25	175	10480	-	-	-	-
PAN18 1&2 Test	25	126	8235	-	-	-	-
C10 Train	10	500	15467	-	-	-	-
C10 Test	10	500	15737	-	-	-	-
C50 Train	50	2500	36845	50.0	3.0	-	-
C50 Test	50	2500	37914	45.6	2.7	-	-
PAN -5	5	600	5551	5.0	3.0	-	-
PAN-10	10	1200	8767	10.0	3.0	-	-
ART-5	5	600	25242	5.0	3.0	-	-
ART-10	10	1200	46228	8.6	2.6	-	-
BLOG-5	5	600	48718	5.0	3.0	-	-
BLOG-10	10	1200	94764	7.1	2.2	-	-
TWE-5	5	600	3794	5.0	3.0	-	-
TWE-10	10	1200	6255	9.6	2.9	-	-
ART+BLOG	10	1200	50666	8.6	2.6	5.0	1.0
ART+TWE	10	1200	27070	8.6	2.6	8.5	1.7
BLOG+TWE	10	1200	30723	9.66	2.9	7.5	1.4
ART+BLOG+TWE	10	1200	36191	10.0	3.0	6.33	1.9

In Table 5.3, the number of authors is followed by a dash (-) symbol attached to the dataset name. PAN11-L and PAN11-S are the names for the PAN large and the PAN small datasets respectively. In this table, the statistics of each dataset include author count, document count, token count and average distributions. Author count is the number of

distinct authors, document count is the number of documents, and token count is the number of distinct tokens in all documents. Missing values are represented by dash (-) symbol due to measurement inconsistencies because of the dataset size and source characteristics. Average (Avg.) values are computed via Eq. (5.1), Eq. (5.2), Eq. (5.3) and Eq. (5.4).

$$Avg.authors\ per\ topic\ = \frac{\sum_{k}^{authors} Number\ of\ topics\ written\ by\ author_k}{Number\ of\ authors} \tag{5.1}$$

$$Avg.topics\ per\ author\ = \frac{\sum_{p}^{topics} number\ of\ authors\ write\ in\ topic_p}{number\ of\ topics} \tag{5.2}$$

$$Avg.authors\ per\ source\ = \frac{\sum_{k}^{authors} Number\ of\ sources\ written\ by\ author_k}{number\ of\ authors} \tag{5.3}$$

$$Avg.sources\ per\ author = \frac{\sum_{s}^{sources} number\ of\ authors\ write\ in\ source_s}{number\ of\ sources} \tag{5.4}$$

In Eq. (5.1) to (5.4), the computation of the general averages for author, topic and source distributions are given. These statistics indicate general properties of the distributions. For example, in Eq. (5.3), when the avg. author per topic is maximum, it can be concluded that each author writes in each topic; however, it cannot be concluded that the documents of authors are distributed uniformly in each topic.

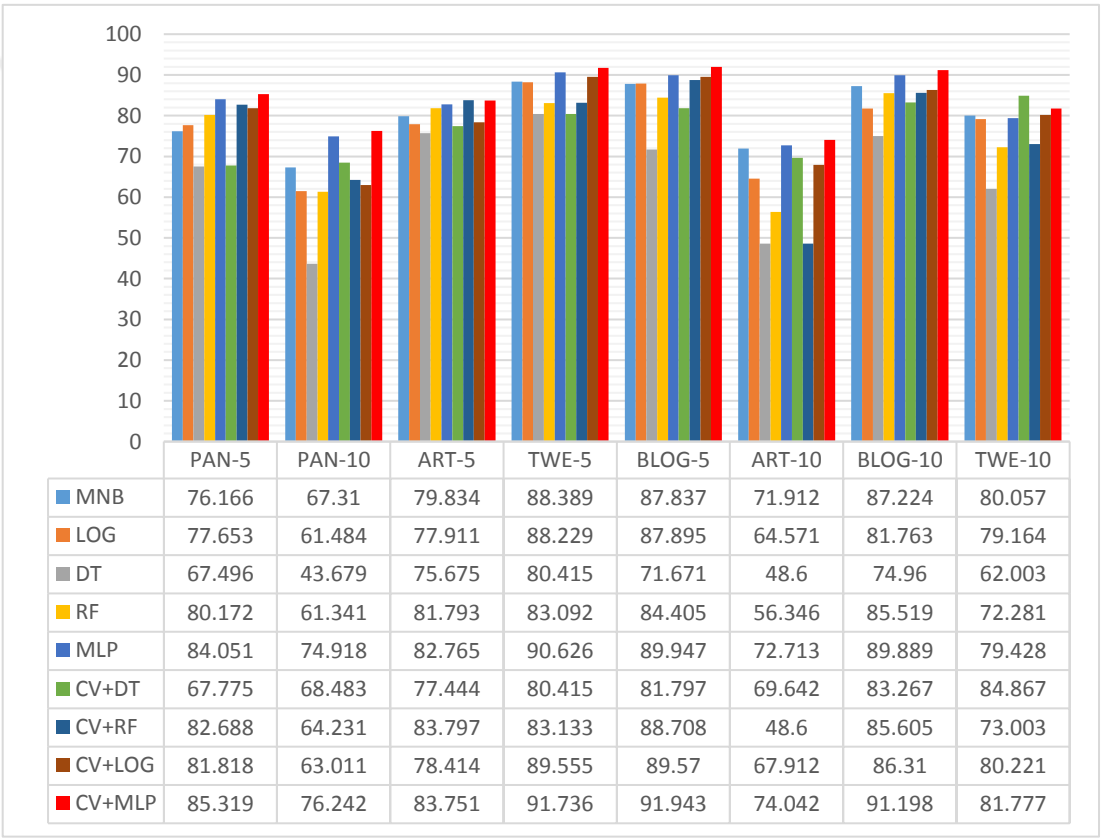
The distributions of topics among authors and author topic tendencies are effective measures in the performance analysis. In order to elaborate topic and source distributions in detail, author specific source and topic distributions for the sampled datasets are given in Figure 5.3, and the competition datasets are excluded from the analysis. In topic analysis, topic models are extracted by latent Dirichlet allocation with a topic size of three on the union of training and testing datasets. These models are later used to analyze the training and testing datasets separately. According to the topic analysis, all datasets have balanced topic distributions, and most of the documents are written on a single topic.

Figure 5.3. *Topic and source distributions for authors in sampled datasets*

In Figure 5.3, each author is analyzed separately for each dataset according to the number of documents in each topic and source. A, T and S are the abbreviations for author, topic and source respectively. Since latent Dirichlet allocation model is used for each dataset separately, the topic distributions of each dataset are unique. In addition to the topic distributions, the source distributions for each author are included for multi-source datasets except Article & Blogs since in the Article & Blogs dataset the authors are writing either in articles or blogs. Source abbreviations are given in datasets in combination order. For example, in Article & Tweets datasets, S0 corresponds to articles and S1 corresponds to tweets. The goal of these visualizations is to understand the topical clusters and distribution of sources in order to test whether a correlation exists with the classification performance.

5.3.3. CV Scaling on term features

In this section, the effects of CV scaling for the term features on single source, mixed source and competition datasets are analyzed. The micro-averaged F-measure scores of the classifiers: multinomial naïve Bayes (MNB), logistic regression (LOG), decision tree (DT), random forest (RF) and multilayer perceptron (MLP) are given in Figure 5.4, Figure 5.5 and Table 5.4. The use of CV scaling prior to the given classifier is shown by + symbol. Since the negative values in document vectors prevent the use of multinomial naïve Bayes classifier, the CV scaling is not applied to this classifier.

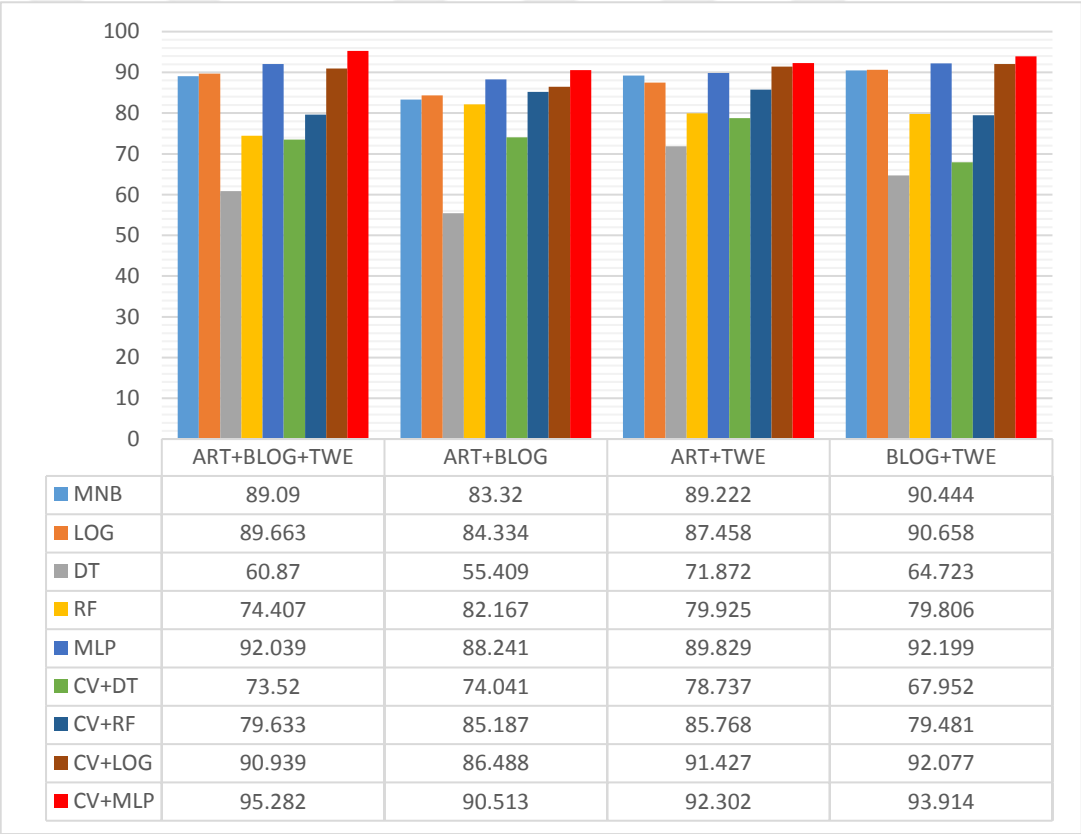


	PAN-5	PAN-10	ART-5	TWE-5	BLOG-5	ART-10	BLOG-10	TWE-10
MNB	76.166	67.31	79.834	88.389	87.837	71.912	87.224	80.057
LOG	77.653	61.484	77.911	88.229	87.895	64.571	81.763	79.164
DT	67.496	43.679	75.675	80.415	71.671	48.6	74.96	62.003
RF	80.172	61.341	81.793	83.092	84.405	56.346	85.519	72.281
MLP	84.051	74.918	82.765	90.626	89.947	72.713	89.889	79.428
CV+DT	67.775	68.483	77.444	80.415	81.797	69.642	83.267	84.867
CV+RF	82.688	64.231	83.797	83.133	88.708	48.6	85.605	73.003
CV+LOG	81.818	63.011	78.414	89.555	89.57	67.912	86.31	80.221
CV+MLP	85.319	76.242	83.751	91.736	91.943	74.042	91.198	81.777

Figure 5.4. *Effects of CV scaling for terms on single source datasets*

Figure 5.4 shows the results of single sources datasets with five and ten authors for pan (PAN), articles (ART), blogs (BLOG) and twitter (TWE) sources. The number of authors in the datasets compiled from these sources is separated by the dash symbol. The datasets with five authors are PAN-5, ART-5, BLOG-5 and TWE-5, and the datasets with ten authors are PAN-10, ART-10, BLOG-10 and TWE-10. The statistical properties of these datasets are given in Section 5.2.2.

In Figure 5.4, the best classification performance without CV scaling is obtained by the MLP classifier. According to these results, introducing the CV scaling improves the performance of MLP, LOG and RF classifiers in all datasets. Exceptionally the performance of the DT classifier is not affected in PAN-5 and TWE-5 datasets. For the other scenarios, the performance improvement of CV scaling is shown to be consistent for each dataset. The performance of CV scaling is dependent on the dataset in terms of the number of documents and character length of the documents. For instance, the CV scaling does not improve the performance of the datasets with five authors as much as that of the datasets with ten authors. In addition to this observation, the improvement on the twitter dataset is also less than other datasets.



	ART+BLOG+TWE	ART+BLOG	ART+TWE	BLOG+TWE
MNB	89.09	83.32	89.222	90.444
LOG	89.663	84.334	87.458	90.658
DT	60.87	55.409	71.872	64.723
RF	74.407	82.167	79.925	79.806
MLP	92.039	88.241	89.829	92.199
CV+DT	73.52	74.041	78.737	67.952
CV+RF	79.633	85.187	85.768	79.481
CV+LOG	90.939	86.488	91.427	92.077
CV+MLP	95.282	90.513	92.302	93.914

Figure 5.5. *Effects of CV scaling for terms on mixed source datasets*

Figure 5.5 shows the results of term features for the datasets compiled with ten authors from the sources of articles (ART), blogs (BLOG) and twitter (TWE) sources. The source combinations include articles, blogs and tweets (ART+BLOG+TWE), articles

and blogs (ART+BLOG), articles and tweets (ART+TWE), and blogs and tweets (BLOG+TWE).

In Figure 5.5, the best performance is obtained by MLP classifier. The improvement of the CV scaling approach is around 3 points for the MLP classifier and is around 10 points for the DT classifier. However, the performance improvement for RF classifier is limited. The maximum improvement for all the classifiers is observed in the ART+BLO+TWE dataset.

Table 5.4. *Effects of CV scaling on term features for competition datasets*

Classifiers	PAN11-S	PAN11-L	C10	C50	PAN18-1	PAN18-2	PAN18 1&2
MNB	61.125	53.288	68.569	65.816	52.449	68.471	39.850
LOG	50.526	39.254	74.236	64.003	52.024	83.022	25.794
DT	43.074	24.261	60.537	33.992	15.553	42.063	20.055
RF	44.200	22.080	73.949	62.117	35.819	68.542	32.821
MLP	64.745	54.506	74.629	67.162	33.677	42.297	31.200
CV+LOG	61.403	52.518	76.127	67.633	55.358	85.879*	47.746*
CV+DT	50.526	24.261	63.011	34.606	16.808	46.031	23.616
CV+RF	44.200	27.070	73.428	63.479	46.323	69.423	35.764
CV+MLP	69.109	59.252	76.198	70.918	40.091	52.721	45.454
DF=5 LOG	52.407	40.037	74.533	66.179	-	-	-
DF=5 CV+LOG	67.449	60.382	76.262*	69.299	-	-	-
DF=5 MLP	61.567	53.078	73.095	65.752	-	-	-
DF=5 CV+MLP	69.825	66.716*	75.651	71.107*	-	-	-
PCA+LOG	57.355	41.403	1.818	56.301	1.796	15.036	0.124
PCA+MLP	60.570	45.276	76.198	50.919	28.022	30.782	13.997
BASELINE	-	-	-	-	55.296	64.253	-
TOP1	71.7*	65.8	-	-	62.500*	67.300	-
TOP2	70.9	64.2	-	-	56.500	77.400	-
TOP3	65.9	59.4	-	-	42.600	58.800	-

Table 5.4 presents CV scaling on C10, C50, PAN11 and PAN18 datasets. The performances of PCA and document frequency selection are also introduced for the LOG and MLP classifiers. The minimum document frequency for filtering the terms is chosen as five, and the dimensionality reduction size for PCA is chosen as 100. DF is not applied on PAN18 datasets due to lack of documents. Similarly, evaluations for top performances and BASELINE are not given for all datasets in the table. The missing evaluation scores are indicated with a dash (-) symbol, and the top scores are indicated with a star (*) symbol. The top performances are given with an asterisk (*). The competition scores are

listed in the table as TOP1 [201], TOP2 [63] and TOP3 [63] for the PAN11 dataset and TOP1 [202], TOP2 [203] and TOP3 [204] for the PAN18 datasets. The top scores of PAN18 dataset are the corresponding problem scores (1 and 2) selected from the best performing approaches in all problems in the PAN 2018 competition. For the PAN18-1 and PAN18-2 datasets, the number of documents is limited; hence, DF filtering was not used. On these datasets, the baseline score (BASELINE) is obtained by character n-grams with the Linear SVM classifier. In PAN18, the training and testing datasets are sampled from distinct domains, and only the authors of the training and testing datasets are common. Therefore, for PAN18 datasets, the common vector is computed separately for the training and testing domains. In combined experiments on the dataset of PAN18 1&2, four domains were used in total by selecting each problem and their training and testing documents as separate domains.

In Table 5.4, the improvements of the CV scaling on MLP and LOG classifiers are between 2 and 5 points. The CV scaling on these classifiers are very competitive to the top performing feature sets on the same datasets. The improvement of the CV scaling is higher than the improvement of the DF, and also applying CV scaling over DF improved the performance for all datasets. On the other hand, the performance of PCA is dependent on the dataset and the classifier. The overall performance of the PCA is worse than the baseline classifier performance. As a matter of fact, the worse performances are obtained by PCA on these datasets.

In PAN18-1 and PAN18-2 datasets, the classification performance of DT, RF and MLP is significantly worse than MNB and LOG classifiers, which indicates that non-linear classifiers are not appropriate for the multi-domain tasks. In these datasets, the best performances are obtained by MNB and LOG classifiers. The improvement of the CV for these classifiers on PAN18-1 and PAN18-2 datasets is higher than the improvement on PAN11 datasets. The most improvement of the CV scaling is observed in PAN18 1&2 dataset. In PAN11-S and PAN11-L datasets, the CV scaling improves the performance of the LOG classifier over DF term selection. In most of these datasets, the CV scaling achieves the highest improvement and the top performance with the MLP classifier.

In Table 5.5, the significance of the scaling technique for the best classification performance in the ART+BLOG+TWE dataset is presented. Significant t-scores are given with a star (*) symbol. For the significance tests, a paired t-test is used on bootstrap-sampled scores for each author.

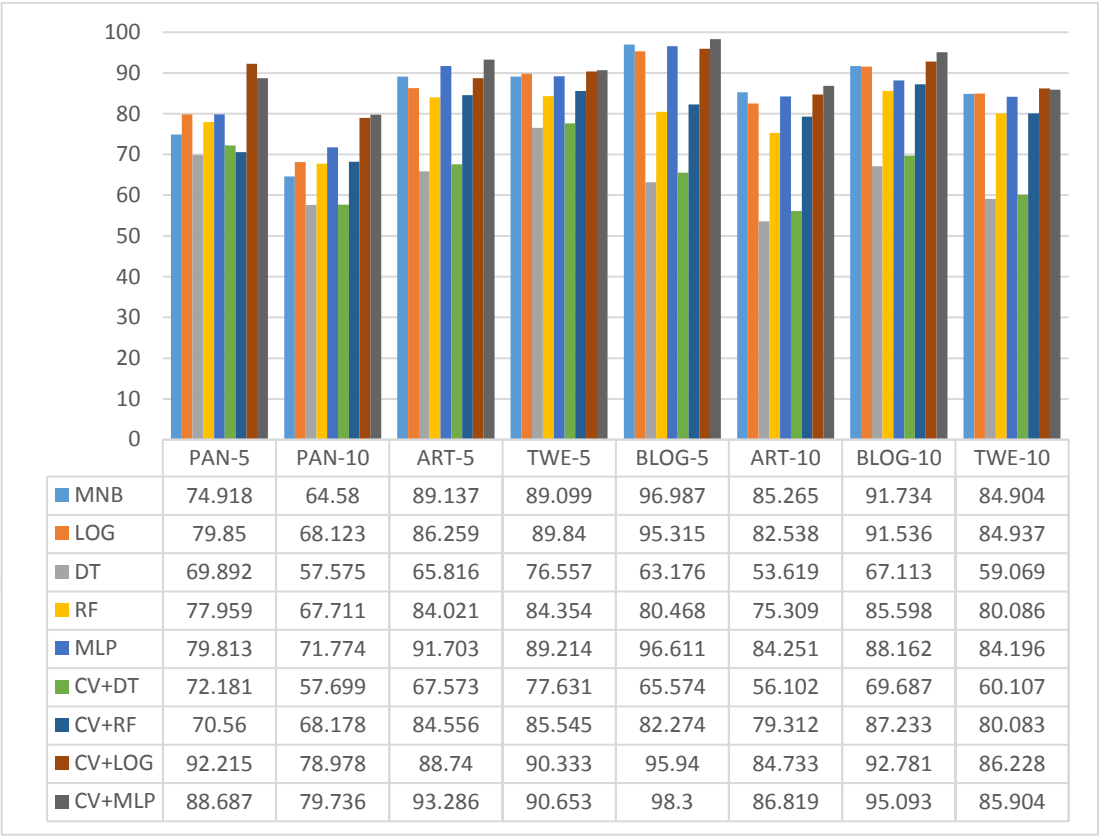
Table 5.5. *Significance tests for term features in the ART+BLOG+TWE dataset*

Test	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
	Mean of F-Measures									
MLP	87.509	91.651	87.272	92.648	89.446	87.753	95.109	89.208	89.741	93.491
CV	94.872	95.218	90.170	95.257	92.490	93.550	95.312	91.745	94.504	97.214
T-Test	E-31*	E-35*	E-7*	E-13*	E-12*	E-26*	0.6409	E-14*	E-15*	E-15*
	Mean of Precisions									
MLP	87.438	89.522	93.108	93.433	93.212	86.401	95.517	92.154	89.780	90.768
CV	93.338	96.049	93.960	94.169	95.038	95.383	94.809	91.816	97.048	97.873
T-Test	E-13*	E-18*	0.3428	0.3998	E-6*	E-24*	0.032*	0.010*	E-25*	E-19
	Mean of Recalls									
MLP	88.708	94.541	83.166	92.458	85.333	90.166	95.041	87.291	90.708	96.916
CV	96.874	94.791	88.749	96.749	92.375	92.375	96.166	93.041	92.624	97.791
T-Test	E-26*	0.061*	E-11*	E-14*	E-18*	E-7*	0.023*	E-16*	E-5*	0.198

In Table 5.5, the author scores are obtained via 20 stratified 10-fold cross-validation experiments, and their averages are given. In bootstrapped sampling, we compute the mean of 20 randomly drawn samples with replacement from 200 experiment scores that are distributed in each fold of 10-fold cross-validation. This process is repeated 1000 times, and a distribution of size 1000 is obtained. After sampling, the equality (H_0) between the distributions of the CV scaling and the MLP classifier is tested using a paired (students) t-test.

5.3.4. CV Scaling on N-gram Features

In this section, the effects of CV scaling for the character n-gram features on single source, mixed source and competition datasets are analyzed. The micro-averaged F-measure scores of the classifiers: multinomial naïve Bayes (MNB), logistic regression (LOG), decision tree (DT), random forest (RF) and multilayer perceptron (MLP) are given in Figure 5.6. The use of CV scaling prior to the given classifier is shown by + symbol. Since the negative values in document vectors prevent the use of multinomial naïve Bayes classifier, the CV scaling is not applied to this classifier.



	PAN-5	PAN-10	ART-5	TWE-5	BLOG-5	ART-10	BLOG-10	TWE-10
MNB	74.918	64.58	89.137	89.099	96.987	85.265	91.734	84.904
LOG	79.85	68.123	86.259	89.84	95.315	82.538	91.536	84.937
DT	69.892	57.575	65.816	76.557	63.176	53.619	67.113	59.069
RF	77.959	67.711	84.021	84.354	80.468	75.309	85.598	80.086
MLP	79.813	71.774	91.703	89.214	96.611	84.251	88.162	84.196
CV+DT	72.181	57.699	67.573	77.631	65.574	56.102	69.687	60.107
CV+RF	70.56	68.178	84.556	85.545	82.274	79.312	87.233	80.083
CV+LOG	92.215	78.978	88.74	90.333	95.94	84.733	92.781	86.228
CV+MLP	88.687	79.736	93.286	90.653	98.3	86.819	95.093	85.904

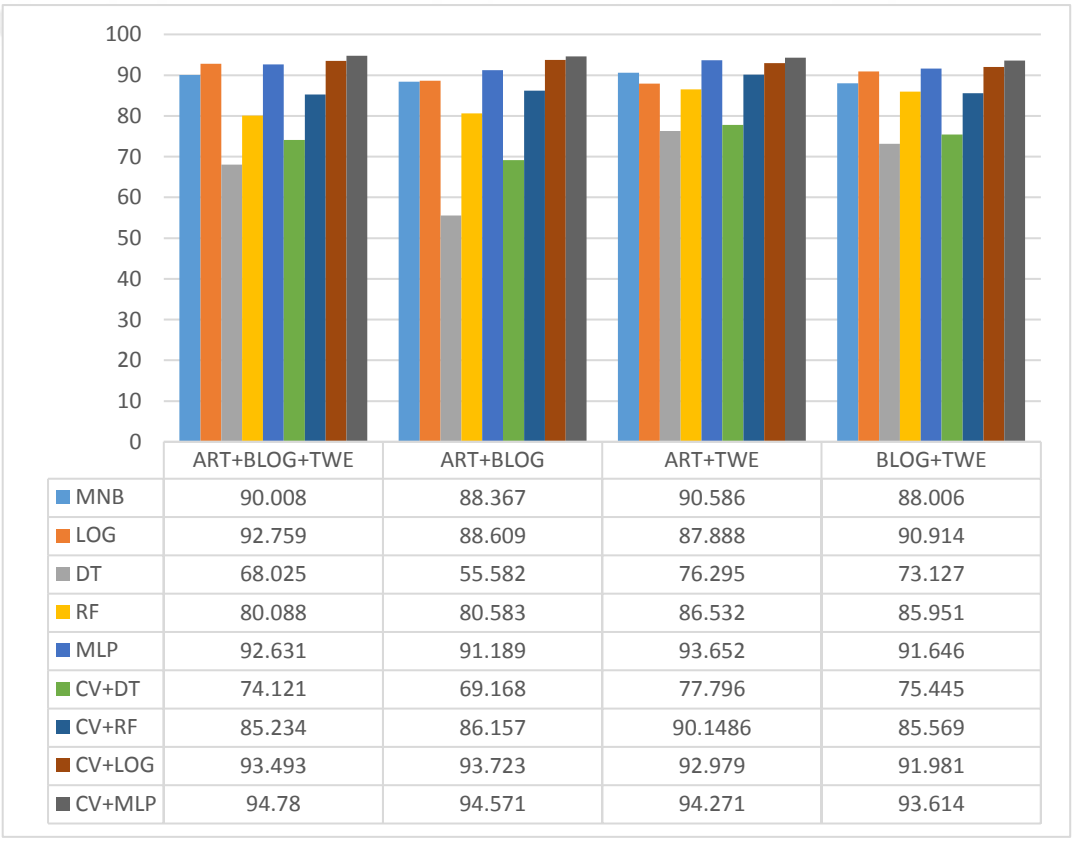
Figure 5.6. *Effects of CV scaling for character n-grams on single source datasets*

Figure 5.6 shows the results of single sources datasets with five and ten authors for pan (PAN), articles (ART), blogs (BLOG) and twitter (TWE) sources. The number of authors in the datasets compiled from these sources is separated by the dash symbol. The datasets with five authors are PAN-5, ART-5, BLOG-5, TWE-5 and the datasets with ten authors are PAN-10, ART-10, BLOG-10 and TWE-10.

In Figure 5.6, the LOG and MLP classifiers show the top performance on character n-gram features for all datasets. For the PAN and TWE datasets, the LOG classifier performance is higher than the MLP classifier. The improvements of the CV scaling on these datasets are observed for DT, LOG and MLP classifiers. The CV scaling has no effect on RF classifier for these datasets. The highest performance improvement is observed in PAN-5 and PAN-10 dataset with LOG classifier. Although the performance improvement is limited in the other datasets, the best performances are always obtained by the CV scaling approach. The improvement of the CV-scaling on MLP classifier is around 6 points for the BLOG-10 dataset and 8 points for PAN-10 dataset.

Furthermore, the results in Figure 5.6 indicate that n-gram features perform better than the term features for the same datasets. Using n-gram features improved the best

performances around 9 points. The performance difference between these feature sets is around 2 points without CV scaling and around 4 points with CV scaling. The improvement of the CV scaling on the n-gram features is higher than the term features. The lowest improvements of the CV scaling is observed as 2 points in ART-5 and ART-10 datasets, and the highest improvements are observed on PAN and BLOG datasets. Based on the analysis given in Figure 5.6, the topic distributions of the ART-5 and ART-10 datasets are similar to BLOG-5 and BLOG-10 datasets, but the average document lengths in BLOG-5 and BLOG-10 datasets are much higher than ART-5 and ART-10 datasets.



	ART+BLOG+TWE	ART+BLOG	ART+TWE	BLOG+TWE
MNB	90.008	88.367	90.586	88.006
LOG	92.759	88.609	87.888	90.914
DT	68.025	55.582	76.295	73.127
RF	80.088	80.583	86.532	85.951
MLP	92.631	91.189	93.652	91.646
CV+DT	74.121	69.168	77.796	75.445
CV+RF	85.234	86.157	90.1486	85.569
CV+LOG	93.493	93.723	92.979	91.981
CV+MLP	94.78	94.571	94.271	93.614

Figure 5.7. *Effects of CV scaling for character n-grams on mixed source datasets*

Figure 5.7 shows the results of n-gram features for the datasets compiled with ten authors from the sources of articles (ART), blogs (BLOG) and twitter (TWE) sources. The source combinations include articles, blogs and tweets (ART+BLOG+TWE), articles and blogs (ART+BLOG), articles and tweets (ART+TWE), and blogs and tweets (BLOG+TWE). The best performances are obtained by MLP classifier for all datasets.

The improvement of the CV scaling over the best performance is around 2 points for the MLP classifier. The best improvement of the CV scaling is around 4 points for DT classifier, and the worst improvement is around 1 point for the MLP classifier.

In Figure 5.7, the CV scaling has consistently improved the performance of each classifier. Similar to the observations on the term features, the improvement of the CV scaling on n-gram features for mixed sources is higher than the improvement for single sources. Moreover, the improvements of the CV scaling on ART+BLOG are higher than the other datasets. In ART+BLOG dataset, the authors are distinct according to the sources, which indicates that the performance improvements of the CV scaling on n-gram features is affected by the diversity of the sources of each author. This case is not observed in the term features. In general, the n-gram features perform better than the term features for the datasets with single sources, but the performance difference between the top performances of the n-gram and the term features is negligible for the datasets with mixed sources.

Table 5.6. *Effects of CV scaling on character n-grams for competition datasets*

Classifiers	PAN11-S	PAN11-L	C10	C50	PAN18-1	PAN18-2	PAN18 1&2
MNB	64.198	58.028	70.463	65.944	51.464	30.158	40.376
LOG	67.356	61.864	76.348	69.534	60.279	59.864	44.851
DT	52.098	38.567	55.881	33.992	32.544	53.319	21.782
RF	44.109	41.846	72.644	60.079	55.539	51.700	42.235
MLP	72.360	70.011	76.434	66.107	49.176	50.918	30.892
CV+LOG	70.205	68.091	77.617	71.287	64.514*	77.922*	57.270
CV+DT	52.081	38.773	55.593	37.012	33.562	63.327	26.087
CV+RF	43.652	38.130	72.584	62.628	55.471	74.277	57.482*
CV+MLP	74.401	71.429*	78.268*	73.675*	50.253	56.462	47.300
DF=5 LOG	68.626	58.346	75.191	69.133	-	-	-
DF=5 CV+LOG	72.789	67.913	77.418	71.964	-	-	-
DF=5 MLP	68.626	69.100	76.314	67.855	-	-	-
DF=5 CV+MLP	74.637*	70.780	77.265	73.432	-	-	-
PCA+LOG	61.929	29.464	1.181	56.636	0.179	0.043	0.002
PCA+MLP	68.877	30.307	76.551	47.093	48.272	49.331	34.730
BASELINE	-	-	-	-	55.296	64.253	-
TOP1	71.7	65.8	-	-	62.500	67.300	-
TOP2	70.9	64.2	-	-	56.500	77.400	-
TOP3	65.9	59.4	-	-	55.200	61.900	-

Table 5.6 presents CV scaling on the C10, C50, PAN11 and PAN18 datasets. The performances of PCA and document frequency selection are also introduced for the LOG and MLP classifiers. The minimum document frequency for filtering the terms is applied as five, and the dimensionality reduction size for PCA is selected 100. DF is not applied on PAN18 datasets due to lack of documents. Similarly, the evaluations for top performances and BASELINE are not given for all the datasets in the table. The missing evaluation scores are indicated by a dash (-) symbol, and the top scores are indicated by a star (*) symbol.

In Table 5.6, the competition scores are listed in the table as TOP1 [201], TOP2 [63] and TOP3 [63] for the PAN11 dataset and TOP1 [202], TOP2 [203] and TOP3 [204] for the PAN18 datasets. The top scores of PAN18 dataset are the corresponding problem scores (1 and 2) selected from the best performing approaches in all the problems in the PAN 2018 competition. For the PAN18-1 and PAN18-2 datasets, the number of documents is limited; hence, DF filtering was not used. On these datasets, the baseline score (BASELINE) is obtained by character n-grams with the Linear SVM classifier. In PAN18, the training and testing datasets are sampled from distinct domains, and only the authors of the training and testing datasets are common. Therefore, for PAN18 datasets, the common vector is computed separately for the training and testing domains. In combined experiments on the dataset of PAN18 1&2, four domains were used in total by selecting each problem and their training and testing documents as separate domains.

In Table 5.6, the improvements of the CV scaling on MLP and LOG classifiers are between 1 and 17 points. The CV scaling on these classifiers is higher than the top performing feature sets on the same datasets. The improvement of the CV scaling is higher than the improvement of the DF, and also applying CV scaling over DF improved the performance for all the datasets. On the other hand, the performance of PCA is dependent on the dataset and the classifier. The overall performance of the PCA is worse than the baseline classifier performance. The worst performances are in fact obtained by PCA on these datasets.

In PAN18-1 and PAN18-2 datasets, the classification performance of DT, RF and MLP is significantly worse than the LOG classifier. In these datasets, the performance of the DT and RF classifiers with character n-gram features is higher than their performance with term features. The improvement of the CV scaling for these classifiers on PAN18-1 and PAN18-2 datasets is higher than the improvements of the CV scaling on PAN11

datasets. The highest improvement of the CV scaling is observed in PAN18 1&2 dataset. On the other hand, in PAN11-S and PAN11-L datasets, the CV scaling achieves the highest improvement and the top performance on the MLP classifier.

6. DISCUSSIONS

In the experimental evaluations, the CV scaling has shown significant improvements for terms and n-gram features in all datasets that have varying document-lengths, genres, topics and sources. In this section, the findings on the evaluations of the CV scaling and the common stylometric approaches are elaborated according to the dimensions of document-lengths, sources and feature sets. The effects of the document length, mix of sources and cross-domain issues on the performance of CV scaling and the feature sets have been provided by the dataset comparisons. In addition to these, the performance of widely applied methods in AA tasks, such as PCA and multilayer perceptron classifier is discussed in this section.

6.1. The Case of Document Length

Traditional AA methods have been successful on literary text documents, blogs and articles. Researchers have suggested a diverse set of features to improve the performance of author classification methods. The case of document length has been separately analyzed for different sets of features, and there is no consensus on the appropriateness of the feature combinations for datasets compiled from documents in different lengths. In this thesis, we report the performance inconsistency of the suggested stylometric features for the short-length and medium-length documents.

In the evaluations, it has been found that the length of the documents does not equally contribute to the performance of all features. The performance improvement of the terms by the increased document-length is much higher than other features. The n-gram features achieved the top performances and performed robustly in varying document-lengths. In short-length documents, the combination of all stylometric features has achieved the top scores, while, in medium-length documents, its performance decreased significantly. On the other hand, the performance improvement of the CV scaling on the same set of features and author sizes suggests that increased document length is beneficial to the performance of the CV scaling, which can also be concluded from the evaluations given in Section 5.3.3 and 5.3.4, where the improvement of CV scaling for BLOG-10 and PAN11-L datasets is much higher than the other datasets.

6.2. The Case of Multiple Sources

In the experimental evaluations, it is observed that the source divergence has a positive impact on both the classification and the CV scaling performance. In combined source experiments, the improvement of the CV scaling over the baseline classification performance has also been increased. The baseline classification performance is indirectly related to the diverse sources since combined sources have a diverse number of topics and diverse vocabulary sizes. In literature, several researchers indicate that topic controlled training with combining multiple topics has increased the performance of AA, which inclines to similar conclusions of the experimental evaluations with topic controlled datasets. Such diversity in the whole dataset encodes useful information into the document vector so that the baseline performance becomes robust.

6.3. The Case of Multilayer Perceptron Classifier

Multilayer perceptron classifier has been very successful in the datasets where training and testing documents are in similar domains. These datasets include all the sampled datasets and the competition datasets except PAN18 datasets. In PAN18 dataset, the performance of MLP classifier is not as high as the performance of other classifiers, which indicates that the MLP is a good candidate classifier for the AA tasks when there are enough documents per author. For the future studies, the MLP classifier should be thoroughly experimented with domain adaptive orientations. Moreover, the improvement of the CV scaling on the MLP dataset is very consistent across all datasets. The performance improvements in single-source and multiple-source datasets are almost constant respectively.

6.4. The Case of Principal Component Analysis

The principal component analysis has been widely applied for both classification and dimensionality reduction in AA tasks. One of the drawbacks of PCA is said to be scalability problems. In the recent studies on multi topic AA tasks, the classification performance of PCA has been questioned. The performance of PCA on a diverse set of features with variations due to topic, document length and domain makes the PCA not a good candidate for heterogeneous datasets. In the experimental evaluations, the effects of PCA on PAN11, C50 and C10 datasets have been compared. Based on these results, it is concluded that PCA is not appropriate for multi topic AA tasks.

6.5. The Case of Domain Adaptation

Traditional classification approaches on AA have studied the effects of new feature sets for multi-topic and multi-genre datasets. Some of these studies have focused on cross-topic and cross-genre issues. However, domain related issues have not been properly studied for author identification since there have been no datasets available. In PAN 2018 AA task, cross-domain datasets have been constructed and experimented. The PAN 2018 datasets have the training and testing document collections from two different domains respectively.

In the experimental evaluations, several classifiers and the CV scaling have been experimented on the dataset. The results indicate that cross-domain prediction of the baseline classifiers has been not as accurate as in single domain experiments. The best performing classifier on other datasets has a very poor performance in this dataset. Linear classifiers, such as MNB and LOG have performed better than DT, RF and MLP. On the other hand, the n-gram feature set has not shown consistent improvement over the term features.

The improvement of the CV scaling has been clearly shown for both n-gram and term features in PAN18 datasets. In the experimental evaluations, in addition to the cross-domain datasets, a multi-cross domain dataset has been constructed by merging the PAN18-1 and PAN18-2 datasets. In this dataset (PAN18 1&2), the n-gram features have been observed to be more robust than term features with all classifiers. Furthermore, the improvement of the CV scaling approach on this dataset is remarkable. It can be concluded that CV scaling is an appropriate approach for cross-domain AA tasks.

7. CONCLUSIONS

Text classification approaches to authorship attribution typically focus on creating effective feature sets for such domains as emails, the Federalist Papers, blogs, comments and newspaper articles. Within this framework, most AA studies focus on identifying useful features. Character n-grams, function words and terms are suggested to be the most effective features within the BOW representation. Moreover, topicality and genre bias create a challenging problem in view of the fact that author's term selection has been shown to be affected by the reader population. Characterization of documents according to source, genre and topic can be formulated as a domain problem. Through this thesis, a novel approach, which outperforms state-of-the-art methods in terms of classification performance for AA in various domains, has been presented. Eliminating the negative effects of the domain for AA by cancelling out correlated patterns or reducing their influence leads to significant improvement with heterogeneous datasets. The common vector approach is used in order to represent domains in a bag-of-words model, and the CV scaling eliminates domain-invariant features from the document. This approach is stochastic, feature-independent and requires one-time computation per domain, which does not burden the classifier during training and prediction.

The experimental evaluations show that the CV scaling outperformed the F-Measure scores of baseline classification performance on sampled and competition datasets. Using the proposed approach on logistic regression and multilayer perceptron classifiers enabled them to surpass their best performance in all domains, and overall, the CV scaling improved the best classification performance of all datasets. Moreover, multi-domain experiments on PAN18 datasets demonstrate that this method is robust and useful for domain adaptation.

One of the major drawbacks of this approach is the quality of the common vectors. Multi-topic documents and short documents, such as tweets are a major problem for the quality of the common vectors. Additional methodological approaches are required to eliminate or reduce inconsistencies within the samples during common vector computation. Further research is needed to increase the representative power of common vectors in different tasks.

REFERENCES

[1] Mendenhall, T. C. (1887). The Characteristic Curves of Composition. *Science (New York, N.Y.)*, *9* (216), 297.

[2] Argamon, S., and Juola, P. (2011). Overview of the International Authorship Identification Competition at PAN-2011. In *CLEF (Notebook Papers/Labs/Workshop)*.

[3] Rocha, A., Scheirery, W. J., Forstall, C. W., Cavalcante, T., Theophilo, A., Shen, B., Carvalho, A. R. B., Stamatatos, E., Scheirer, W. J., Forstall, C. W., Cavalcante, T., Theophilo, A., Shen, B., … Stamatatos, E. (2016). Authorship attribution for social media forensics. In *IEEE Transactions on Information Forensics and Security* Vol. PP, 5–33.

[4] Abbasi, A., and Chen, H. (2008). Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems*, *26* (2), 7:1-7:29. doi:10.1145/1344411.1344413

[5] Argamon-Engelson, S., Koppel, M., and Avneri, G. (1998). Style-based text categorization: What newspaper am I reading. In *Proc. of the AAAI Workshop on Text Categorization* 1–4.

[6] Sari, Y., and Stevenson, M. (2015). A Machine Learning-based Intrinsic Method for Cross-topic and Cross-genre Authorship Verification. *CLEF (Working Notes)*.

[7] Seroussi, Y., Bohnert, F., and Zukerman, I. (2014). Authorship Attribution with Author-aware Topic Models. *Computational Linguistics*, *40* (2), 269–310.

[8] Sboev, A., Litvinova, T., Gudovskikh, D., Rybka, R., and Moloshnikov, I. (2016). Machine Learning Models of Text Categorization by Author Gender Using Topic-independent Features. *Procedia Computer Science*, *101*, 135–142.

[9] Sapkota, U., Solorio, T., Montes, M., Bethard, S., and Rosso, P. (2014). Cross-Topic Authorship Attribution: Will Out-Of-Topic Data Help? In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* 1228–1237.

[10] Mikros, G. K., and Argiri, E. K. (2007). Investigating topic influence in authorship attribution. In *CEUR Workshop Proceedings* Vol. 276, 29–35.

[11] Tschuggnall, M., and Specht, G. (2014). Enhancing Authorship Attribution By Utilizing Syntax Tree Profiles. *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, 195–199.

[12] Hitschler, J., Berg, E. Van Den, and Rehbein, I. (2017). Authorship Attribution with Convolutional Neural Networks and POS-Eliding. *Association for Computational Linguistics*, (2013), 53–58.

[13] Safder, I., Shabbir, M., Paper, C., and Itu, M. I. (2017). Deep Stylometry and Lexical & Syntactic Features Based Author Attribution on PLoS Digital Repository. In *Digital Libraries: Data, Information, and Knowledge for Digital Lives: 19th International Conference on Asia-Pacific Digital Libraries, ICADL 2017, Bangkok, Thailand, November 13-15, 2017, Proceedings* Vol. 10647, 119.

[14] Bartoli, A., Dagri, A., De Lorenzo, A., Medvet, E., Tarlao, F., Lorenzo, A. De, Medvet, E., and Tarlao, F. (2015). An Author Verification Approach Based on Differential Features. *CEUR WORKSHOP PROCEEDINGS*, *1391*.

[15] Kaster, A., Siersdorfer, S., and Weikum, G. (2005). Combining text and linguistic document representations for authorship attribution. In *SIGIR workshop: stylistic analysis of text for information access* Vol. 1, 27–35.

[16] Ruseti, S., and Rebedea, T. (2012). Authorship Identification Using a Reduced Set of Linguistic Features. In *CLEF (Online Working Notes/Labs/Workshop)* 4–7.

[17] Tanguy, L., Sajous, F., Calderone, B., and Hathout, N. (2012). Authorship Attribution: Using Rich Linguistic Features when Training Data is Scarce. In *PAN Lab at CLEF*.

[18] Tanguy, L., Urieli, A., and Calderone, B. (2011). A multitude of linguistically- rich features for authorship attribution. In *PAN Lab at CLEF*.

[19] Stamatatos, E., Daelemans, W., Verhoeven, B., Juola, P., López-López, A., Potthast, M., and Stein, B. (2015). Overview of the Author Identification Task at PAN 2015. In *CLEF 2015 Evaluation Labs and Workshop Working Notes Papers*. Toulouse, France: CEUR.

[20] Overdorf, R., and Greenstadt, R. (2016). Blogs, Twitter Feeds, and Reddit Comments: Cross-domain Authorship Attribution. *Proceedings on Privacy Enhancing Technologies*, *2016* (3), 155–171.

[21] Stamatatos, E. (2017). Authorship attribution using text distortion. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers* Vol. 1, 1138–1149.

[22] Stamatatos, E. (2018). Masking topic-related information to enhance authorship attribution. *Journal of the Association for Information Science and Technology*, *69* (3), 461–473.

[23] De Vel, O., Anderson, A., Corney, M. W., and Mohay, G. (2001). Multi-Topic E-mail Authorship Attribution Forensics. *Proceedings of ACM Conference on Computer SecurityWorkshop on Data Mining for Security Applications*, 8 p.

[24] Halvani, O., Winter, C., and Graner, L. (2017). Authorship verification based on compression-models. *arXiv preprint arXiv:1706.00516*, 1–17. Retrieved from http://arxiv.org/abs/1706.00516

[25] Savoy, J. (2013). Comparative evaluation of term selection functions for authorship attribution. *Digital Scholarship in the Humanities*, *30* (2), 246–261.

[26] Forstall, C., and Scheirer, W. (2010). Features From Frequency: Authorship and Stylistic Analysis Using Repetitive Sound. *Journal of the Chicago Colloquium on Digital Humanities and Computer Science*, *1* (2).

[27] Hayes, J. H. (2008). Authorship Attribution: A Principal Component and Linear Discriminant Analysis of the Consistent Programmer Hypothesis. *International Journal of Computers and Their Applications*, *15* (2), 79–99.

[28] Markov, I., Stamatatos, E., and Sidorov, G. (2017). Improving cross-topic authorship attribution: The role of pre-processing. In *Proceedings of the 18th International Conference on Computational Linguistics and Intelligent Text Processing. CICLing*.

[29] Stamatatos, E. (2013). On the Robustness of Authorship Attribution Based on Character N-Gram Features. *Journal of Law & Policy*, *2018* (January 2013), 421–439.

[30] Gómez-Adorno, H., Posadas-Durán, J.-P., Sidorov, G., and Pinto, D. (2018). Document embeddings learned on various types of n-grams for cross-topic authorship attribution. *Computing*.

[31] Houvardas, J., and Stamatatos, E. (2006). N-gram feature selection for authorship identification. *Artificial Intelligence Methodology Systems and Applications*, *4183*, 77–86.

[32] Schwartz, M. B. (2016). An Examination of Cross-Domain Authorship Attribution Techniques. *CUNY Academic Works*.

[33] Schein, A. I., Caver, J. F., Honaker, R. J., and Martell, C. H. (2010). Author Attribution Evaluation with Novel Topic Cross-validation. In *KDIR* 206–215.

[34] Sapkota, U. (2015). *Improving the performance of cross-domain authorship attribution*. The University of Alabama at Birmingham.

[35] Baayen, H., Halteren, H. Van, Neijt, A., and Tweedie, F. (2002). An experiment in authorship attribution. *6th JADT*, *I* (January), 69–75.

[36] Savoy, J. (2012). Authorship Attribution: A Comparative Study of Three Text Corpora and Three Languages. *Journal of Quantitative Linguistics*, *19* (2), 132–161.

[37] Binongo, J. N. G., and Smith, M. W. A. (1999). The application of principal component analysis to stylometry. *Literary and Linguistic Computing*, *14* (4), 445–466.

[38] Craig, H., and Kinney, A. F. (2009). Methods. In H. Craig & A. F. E. Kinney (Eds.), *Shakespeare, Computers, and the Mystery of Authorship* 15–39. Cambridge University Press.

[39] Tamura, A., and Zhao, Q. (2007). Rough common vector: A new approach to face recognition. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, (2), 2366–2371.

[40] Jin, S., Nam, M., and Han, S.-T. (2006). Improvements on Common Vector Approach Using k -Clustering Method. In *Pacific Rim Knowledge Acquisition Workshop* 199–206.

[41] Wen, Y., Lu, Y., Shi, P., and Wang, P. S. P. (2011). Common Vector Based Face Recognition Algorithm. *Pattern Recognition, Machine Intelligence and Biometrics*.

[42] Edizkan, R., Gülmezo, M. B., Ergin, S., and Barkana, A. (2005). Improvements on common vector approach for multi class problems. *Signal Processing Conference, IEEE*, *13th Europ*, 1–4.

[43] He, Y., Zhao, L., and Zou, C. (2006). Face recognition using common faces method. *Pattern Recognition*, *39* (11), 2218–2222.

[44] Gulmezoglu, M. B., Dzhafarov, V., Keskin, M., and Barkana, a. (1999). A novel approach to isolated word recognition. *IEEE Transactions on Speech and Audio Processing*, *7* (6), 620–628.

[45] Peng, R. D., and N, W. H. (2002). Quantitative analysis of literary style. *The American Statistician*, *56* (3), 175–185.

[46] Holmes, D. I., and Crofts, D. W. (2010). The diary of a public man: a case study in traditional and non-traditional authorship attribution. *Literary and Linguistic Computing*, *25* (2), 179–197.

[47] Module, C. F. D. (1921). *The problem of the Pastoral Epistles*. Oxford University Press.

[48] Mosteller, F., and Wallace, D. L. (1963). Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed Federalist Papers. *Journal of the American Statistical Association*, *58* (302), 275–309.

[49] Williams, A. C. B., Biometrika, S., Mar, N., and Williams, C. B. (1940). A note on the statistical analysis of sentence-length as a criterion of literary style. *Biometrika*, *31* (3/4), 356–361.

[50] Yule, C. U., Greg, W. W., and Yule, G. (1944). *The statistical study of literary vocabulary*. *The Modern Language Review* Vol. 39. Cambridge University Press.

[51] YULE, G. U. (1939). On sentence-length as a statistical characteristic of style in prose: With application to two cases of disputed authorship. *Biometrika*, *30* (3/4), 363–390.

[52] Sarndal, C.-E. (1967). On deciding cases of disputed authorship. *Applied Statistics*, *16* (3), 251–268.

[53] Brinegar, C. S. (1963). Mark Twain and the Quintus Curtius Snodgrass letters: A statistical test of authorship. *Journal of the American statistical Association*, *58* (301), 85–96.

[54] Holmes, D. I., and Forsyth, R. S. (1995). The Federalist Revisited: New Directions in Authorship Attribution. *Literary and Linguistic Computing*, *10* (2), 111–127.

[55] Ledger, G., and Merriam, T. (1994). Shakespeare, Fletcher, and the two noble Kinsmen. *Literary and Linguistic Computing*, *9* (3), 235–248.

[56] Kjell, B., Addison Woods, W., Frieder, O., Woods, W. A., and Frieder, O. (1995). Information retrieval using letter tuples with neural network and nearest neighbor classifiers. In *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on* Vol. 2, 1222–1226.

[57] Holmes, D. I., Robertson, M., and Paez, R. (2001). Stephen Crane and the New York Tribune: A case study in traditional and non-traditional authorship attribution. *Computers and the Humanities*, *35* (1964), 315–331.

[58] Binongo, J. N. G. (2003). Who Wrote the 15th Book of Oz? An Application of Multivariate Analysis to Authorship Attribution. *Chance*, *16* (2), 9–17.

[59] Argamon-Engelson, S., Koppel, M., and Avneri, G. (1998). Style-based text categorization: What newspaper am I reading? *Proceedings of AAAI Workshop on Learning for Text Categorization*, 1–4.

[60] Stamatatos, E., Fakotakis, N., and Kokkinakis, G. (2000). Automatic Text Categorization in Terms of Genre and Author. *Computational Linguistics*, *26* (4), 471–495.

[61] Ramyaa, C. H. C., Rasheed, K., and He, C. (2004). Using machine learning techniques for stylometry. In *Proceedings of International Conference on Machine Learning*.

[62] Pavelec, D., Justino, E., and Oliveira, L. S. (2007). Author identification using stylometric features. *Inteligencia Artificial*, *11* (36), 59–65.

[63] Tanguy, L., Urieli, A., Calderone, B., Hathout, N., and Sajous, F. (2011). A multitude of linguistically-rich features for authorship attribution. In *PAN Lab at CLEF*.

[64] Chaski, C. E. (2005). Who's At The Keyboard? Authorship Attribution in Digital Evidence Investigations. *International Journal of Digital Evidence*, *4* (1), 1–13.

[65] Nieto, V. (2004). Authorship attribution with help of language engineering. *Methodology*, 1–5.

[66] Björklund, J., and Zechner, N. (2017). Syntactic methods for topic-independent authorship attribution. *Natural Language Engineering*, *23* (5), 789–806.

[67] Luyckx, K. (2004). Syntax-Based Features and Machine Learning techniques for Authorship Attribution. *Cntsuaacbe*, 182.

[68] Halvani, O., Winter, C., and Pflug, A. (2016). Authorship verification for different languages, genres and topics. *Digital Investigation*, *16*, S33–S43.

[69] Kestemont, M., Luyckx, K., Daelemans, W., and Crombez, T. (2012). Cross-Genre Authorship Verification Using Unmasking. *English Studies*, *93* (3), 340–356.

[70] Ashraf, S., Iqbal, H. R., and Nawab, R. M. A. (2016). Cross-Genre Author Profile Prediction Using Stylometry-Based Approach. In *CLEF (Working Notes)* 992–999.

[71] Eder, M. (2010). Does Size Matter? Authorship Attribution, Small Samples, Big Problem. *Proceedings of Digital Humanities*, 132–135.

[72] Madigan, D., Genkin, A., Lewis, D. D., Argamon, S., Fradkin, D., and Ye, L. (2005). Author Identification on the Large Scale. In *Proc. of the Meeting of the Classification Society of North America* Vol. 13.

[73] Layton, R., Watters, P., and Dazeley, R. (2010). Authorship attribution for Twitter in 140 characters or less. *2nd Cybercrime and Trustworthy Computing Workshop, CTC 2010*, 1–8.

[74] Chung, C. K., and Pennebaker, J. W. (2007). The psychological function of function words. *Social Communication: Frontiers of Social Psychology*, (January), 343–359.

[75] Kestemont, M. (2014). Function Words in Authorship Attribution From Black Magic to Theory ? *3rd Workshop on Computational Linguistics for Literature (CLfL 2014)*, 59–66.

[76] Sapkota, U., Bethard, S., y Gómez, M. M., and Solorio, T. (2015). Not all character n-grams are created equal: A study in authorship attribution. *2015 Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL HLT 2015)*, 93–102.

[77] Mikros, G., and Perifanos, K. (2013). Authorship Attribution in Greek Tweets Using Author's Multilevel N-gram Profiles. *2013 AAAI Spring Symposium Series*, 17–23.

[78] Zhao, Y., and Zobel, J. (2005). Effective and Scalable Authorship Attribution Using Function Words, 174–189.

[79] Markov, I., Baptista, J., and Pichardo-Lagunas, O. (2017). Authorship attribution in portuguese using character N-grams. *Acta Polytechnica Hungarica*, *14* (3), 59–78.

[80] Sanchez-Perez, M. A., Markov, I., Gómez-Adorno, H., and Sidorov, G. (2017). Comparison of Character n-grams and Lexical Features on Author, Gender, and Language Variety Identification on the Same Spanish News Corpus. In *International Conference of the Cross-Language Evaluation Forum for European Languages* Vol. 10456 LNCS, 145–151.

[81] Matthews, R. A. J., and Merniam, T. V. N. (1993). Neural Computation in Stylometry I: An Application to the Works of Shakespeare and Fletcher. *Literary and Linguistic Computing*, *8* (4), 203–9.

[82] Holmes, D. I., and Kardos, J. (2003). Who Was the Author? An Introduction to Stylometry. *Chance*, *16* (2), 5–8.

[83] Koppel, M., and Winter, Y. (2014). Determining if Two Documents are by the Same Author, 1–35.

[84] Koppel, M., Schler, J., and Argamon, S. (2011). Authorship attribution in the wild. *Language Resources and Evaluation*, *45* (1), 83–94.

[85] Potha, N., and Stamatatos, E. (2017). An improved impostors method for authorship verification. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10456 LNCS*, 138–144.

[86] Ruder, S., Ghaffari, P., and Breslin, J. G. (2016). Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. *arXiv preprint arXiv:1609.06686*. Retrieved from http://arxiv.org/abs/1609.06686

[87] Barbon, S., Igawa, R. A., and Bogaz Zarpelão, B. (2017). Authorship verification applied to detection of compromised accounts on online social networks: A continuous approach. *Multimedia Tools and Applications*, *76* (3), 3213–3233.

[88] Sapkota, U., Solorio, T., Montes, M., and Bethard, S. (2016). Domain Adaptation for Authorship Attribution: Improved Structural Correspondence Learning. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2226–2235. doi:10.18653/v1/P16-1210

[89] Petmanson, T. (2013). Authorship identification using correlations of frequent features. In *Proceedings of CLEF*.

[90] Wang, S., Ferracane, E., and Mooney, R. J. (2017). Leveraging Discourse Information Effectively for Authorship Attribution. *arXiv preprint arXiv:1709.02271*.

[91] Soler, J., and Wanner, L. (2017). On the Relevance of Syntactic and Discourse Features for Author Profiling and Identification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, *2*, 681–687. doi:10.18653/v1/E17-2108

[92] Feng, V. W., and Hirst, G. (2013). Authorship Verification with Entity Coherence and Other Rich Linguistic Features. In *Proceedings of CLEF* Vol. 13, 5–8.

[93] Luyckx, K., Daelemans, W., Hamilton, A., and Madison, J. (2008). Authorship Attribution and Verification with Many Authors and Limited Data. *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, (August), 513–520.

[94] Castro, D., Adame, Y., Pelaez, M., and Mu??oz, R. (2015). Authorship verification, combining linguistic features and different similarity functions. *CEUR Workshop Proceedings*, *1391* (September).

[95] Solorio, T., and Pillay, S. (2011). Authorship Identification with Modality Specific Meta Features - Notebook for PAN at CLEF 2011. *Working Notes Papers of the CLEF 2011 Evaluation Labs*, 1–11.

[96] Argamon, S., Whitelaw, C., Chase, P., Hota, S. R., Garg, N., and Levitan, S. (2007). Stylistic text classification using functional lexical features: Research Articles. *Journal of the American Society for Information Science and Technology*, *58* (6), 802–822.

[97] Sapkota, U., and Solorio, T. (2012). Sub-Profiling by Linguistic Dimensions to Solve the Authorship Attribution Task. *Working Notes Papers of the CLEF 2012 Evaluation Labs*.

[98] Van Halteren, H. (2004). Linguistic profiling for author recognition and verification. *Linguistic Profiling for Author Recogintion And Verification*, *30*, 199–es.

[99] Agun, H. V., Yilmazel, S., and Yilmazel, O. (2017). Effects of language processing in Turkish authorship attribution. *2017 IEEE International Conference on Big Data (Big Data)*, 1876–1881.

[100] Solorio, T., Pillay, S., Raghavan, S., and Montes-y-Gómez, M. (2011). Modality Specific Meta Features for Authorship Attribution in Web Forum Posts. *Proceedings of the 5th International Joint Conference on Natural Language Processing*, *156*, 156–164.

[101] STAMATATOS, E. (2006). Authorship Attribution Based on Feature Set Subspacing Ensembles. *International Journal on Artificial Intelligence Tools*, *15* (05), 823–838.

[102] Van Cranenburgh, A. (2012). Literary authorship attribution with phrase-structure fragments. In *Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature* 59–63.

[103] Savoy, J. (2013). Feature selections for authorship attribution. *Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13*, 939.

[104] Savoy, J. (2015). Comparative evaluation of term selection functions for authorship attribution. *Digital Scholarship in the Humanities*, *30* (2), 246–261.

[105] Yavanoglu, O. (2016). Intelligent Authorship Identification with using Turkish Newspapers Metadata. In *2016 IEEE International Conference on Big Data* 1895–1900.

[106] Azam, N., and Yao, J. (2012). Comparison of term frequency and document frequency based feature selection metrics in text categorization. *Expert Systems with Applications*, *39* (5), 4760–4768.

[107] Jankowska, M., and Milios, E. (2013). Proximity Based One-Class Classification with Common N-Gram Dissimilarity for Authorship Verification Task - Notebook for PAN at CLEF 2013. *Working Notes Papers of the CLEF 2013 Evaluation Labs*.

[108] van Dam, M., and Hauff, C. (2014). Large-scale author verification: Temporal and Topical Influence. *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval - SIGIR '14*, 1039–1042.

[109] Jankowska, M., Kešelj, V., and Milios, E. (2013). CNG text classification for authorship profiling task Vol. 1, 2–4.

[110] Koppel, M., Akiva, N., and Dagan, I. (2006). Feature instability as a criterion for selecting potential style markers. *Journal of the American Society for Information Science and Technology*, *57* (11), 1519–1525.

[111] Holmes, D. I., and Forsyth, R. S. (1995). The Federalist Revisited: New Directions in Authorship Attribution. *Literary and Linguistic Computing*, *10* (2), 111–127.

[112] Rudman, J. (1998). The State of Authorship Attribution Studies: Some Problems and Solutions. *Computers and the Humanities*, *31*, 351–365.

[113] Savoy, J. (2012). Authorship attribution: A comparative study of three text corpora and three languages. *Journal of Quantitative Linguistics*, *19* (2), 132–161.

[114] Zhang, C., Wu, X., Niu, Z., and Ding, W. (2014). Authorship identification from unstructured texts. *Knowledge-Based Systems*, *66*, 99–111.

[115] Kestemont, M., Moens, S., and Deploige, J. (2015). Collaborative authorship in the twelfth century: A stylometric study of hildegard of bingen and guibert of gembloux. *Digital Scholarship in the Humanities*, *30* (2), 199–224.

[116] Le, Q., and Mikolov, T. (2014). Distributed Representations of Sentences and Documents. *International Conference on Machine Learning - ICML 2014*, *32*, 1188–1196.

[117] Socher, R., and Manning, C. (2013). Deep Learning for NLP (without magic). *Naacl 2013*, 1–204.

[118] Johnson, R., and Zhang, T. (2015). Semi-supervised Convolutional Neural Networks for Text Categorization via Region Embedding. *Nips*, 1–9.

[119] Ge, Z., and Sun, Y. (2016). Domain specific author attribution based on feedforward neural network language models. *arXiv preprint arXiv:1602.07393*. Retrieved from http://arxiv.org/abs/1602.07393

[120] Alsulami, B., Dauber, E., Harang, R., Mancoridis, S., and Greenstadt, R. (2017). Source Code Authorship Attribution Using Long Short-Term Memory Based Networks. In *European Symposium on Research in Computer Security* Vol. 10492, 65–82.

[121] Bagnall, D. (2015). Author identification using multi-headed recurrent neural networks. *arXiv preprint arXiv:1506.04891*, *1391*.

[122] Song, Y., and Lee, C.-J. (2017). Learning User Embeddings from Emails. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers* Vol. 2, 733–738.

[123] Shrestha, P., Sierra, S., González, F. A., Rosso, P., Montes-Y-Gómez, M., Solorio, T., Gonzalez, F., Montes, M., Rosso, P., and Solorio, T. (2017). Convolutional neural networks for authorship attribution of short texts. *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of Conference*, *2*, 669–674.

[124] Brocardo, M. L., Traore, I., Woungang, I., and Obaidat, M. S. (2017). Authorship verification using deep belief network systems. *International Journal of Communication Systems*, *30* (12).

[125] Ding, S. H. H., Fung, B. C. M., Iqbal, F., and Cheung, W. K. (2019). Learning Stylometric Representations for Authorship Analysis. *IEEE Transactions on Cybernetics*, *49* (1), 107–121.

[126] Dai, A. M., Olah, C., Le, Q. V., and Corrado, G. S. (2014). Document Embedding with Paragraph Vectors. *NIPS WS on deep neural networks and representation learning*, 1–9.

[127] Kenter, T., and Rijke, M. de. (2015). Short Text Similarity with Word Embeddings Categories and Subject Descriptors. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM 2015)*, 1411–1420.

[128] Li, J., and Jurafsky, D. (2015). Do Multi-Sense Embeddings Improve Natural Language Understanding? *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (September), 1722–1732.

[129] Huang, C., Qiu, X., and Huang, X. (2014). Text classification with document embeddings. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data* 131–140. Springer.

[130] Agun, H. V., and Yilmazel, O. (2017). Document embedding approach for efficient authorship attribution. In *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)* 194–198. IEEE.

[131] Markov, I., and Helena, G. (n.d.). Author Profiling with doc2vec Neural Network-Based Document Embeddings.

[132] Gómez-Adorno, H., Markov, I., Sidorov, G., Posadas-Durán, J. P., Sanchez-Perez, M. A., and Chanona-Hernandez, L. (2016). Improving Feature Representation Based on a Neural Network for Author Profiling in Social Media Texts. *Computational Intelligence and Neuroscience*, *2016*.

[133] Kocher, M., and Savoy, J. (2018). Distributed language representation for authorship attribution. *Digital Scholarship in the Humanities*, *33* (2), 425–441.

[134] Okuno, S., Asai, H., and Yamana, H. (2015). A challenge of authorship identification for ten-thousand-scale microblog users. *Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014*, 52–54.

[135] Luyckx. (2010). Scalability issues in authorship attribution.

[136] Mikros, G. K. G., and Perifanos, K. (2011). Authorship identification in large email collections : Experiments using features that belong to different linguistic levels. In *Notebook for PAN at CLEF*.

[137] Luyckx, K., and Daelemans, W. (2011). The effect of author set size and data size in authorship attribution. *Literary and Linguistic Computing*, *26* (1), 35–55.

[138] Flekova, L., and Gurevych, I. (2013). Can we hide in the web? Large scale simultaneous age and gender author profiling in social media: Notebook for PAN at CLEF 2013. *CEUR Workshop Proceedings*, *1179*.

[139] Shrestha, P., Mukherjee, A., and Solorio, T. (2016). Large Scale Authorship Attribution of Online Reviews. In *International Conference on Intelligent Text Processing and Computational Linguistics* 221–232.

[140] Luyckx, K., and Daelemans, W. (2011). The effect of author set size and data size in authorship attribution. *Literary and Linguistic Computing*, *26* (1), 35–55.

[141] Brocardo, M. L., Traore, I., Saad, S., and Woungang, I. (2013). Authorship verification for short messages using stylometry. *2013 International Conference on Computer, Information and Telecommunication Systems, CITS 2013*.

[142] Zhao, Y. (2007). *Effective Authorship Attribution in Large Document Collections*. RMIT University.

[143] Evert, S., Proisl, T., Jannidis, F., Reger, I., Pielström, S., Schöch, C., and Vitt, T. (2017). Understanding and explaining Delta measures for authorship attribution. *Digital Scholarship in the Humanities*, *32* (June 2017), ii4-ii16.

[144] Burrows, J. F. (1989). 'An ocean where each kind...': Statistical analysis and some major determinants of literary style. *Computers and the Humanities*, *23* (4–5), 309–321.

[145] Cerra, D., Datcu, M., and Reinartz, P. (2014). Authorship analysis based on data compression. *Pattern Recognition Letters*, *42* (1), 79–84.

[146] Lambers, M., and Veenman, C. J. (2009). Forensic authorship attribution using compression distances to prototypes. In *International Workshop on Computational Forensics* Vol. 5718 LNCS, 13–24.

[147] Li, Z. (2013). An Exploratory Study on Authorship Verification Models for Forensic Purpose.

[148] Klimt, B., and Yang, Y. (2004). The Enron Corpus: A New Dataset for Email Classification Research, 217–226.

[149] Potthast, M., Braun, S., Buz, T., Duffhauss, F., Friedrich, F., Gülzow, J. M., Köhler, J., Lötzsch, W., Müller, F., Müller, M. E., Paßmann, R., Reinke, B., Rettenmeier, L., … Silvello, G. (2016). Who Wrote the Web? Revisiting Influential Author Identification Research Applicable to Information Retrieval. In N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, G. M. Di Nunzio, … G. Silvello (Eds.), *Advances in Information Retrieval. 38th European Conference on IR Research (ECIR 16)* Vol. 9626, 393–407. Berlin Heidelberg New York: Springer.

[150] Mechti, S., Jaoua, M., Belguith, L. H., and Faiz, R. (2014). Machine learning for classifying authors of anonymous tweets , blogs , reviews and social media. In *Proceedings of the PAN@ CLEF, Sheffield, England* 1137–1142.

[151] Almishari, M., Kaafar, D., Oguz, E., and Tsudik, G. (2014). Stylometric Linkability of Tweets. *Proceedings of the 13th Workshop on Privacy in the Electronic Society - WPES '14*, 205–208.

[152] Albadarneh, J., Talafha, B., Al-Ayyoub, M., Zaqaibeh, B., Al-Smadi, M., Jararweh, Y., and Benkhelifa, E. (2015). Using Big Data Analytics for Authorship Authentication of Arabic Tweets. *Proceedings - 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing, UCC 2015*, (December), 448–452.

[153] Sultana, M., Polash, P., Gavrilova, M., Paul, P. P., Gavrilova, M., Polash, P., and Gavrilova, M. (2017). Authorship recognition of tweets: A comparison between social behavior and linguistic profiles. In *2017 IEEE International Conference on Systems, Man, and Cybernetics* 471–476. IEEE.

[154] Hong, R., Tan, R., and Tsai, F. S. (2010). Authorship identification for online text. *Proceedings - 2010 International Conference on Cyberworlds, CW 2010*, 155–162.

[155] Afroz, S., Caliskan-Islam, A., Stolerman, A., Greenstadt, R., and McCoy, D. (2014). Doppelgänger finder: Taking stylometry to the underground. *Proceedings - IEEE Symposium on Security and Privacy*, 212–226. doi:10.1109/SP.2014.21

[156] Goldstein-Stewart, J., Winder, R., and Sabin, R. E. (2009). Person identification from text and speech genre samples. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics* 336–344.

[157] Luyckx, K., and Daelemans, W. (2005). Shallow Text Analysis and Machine Learning for Authorship Attribution. *Technology*, *4*, 149–160.

[158] Diederich, J., Kinderman, J., Leopold, E., and Paass, G. (2003). Authorship Attribution with Support Vector Machines. *Applied Intelligence*, *19* (1), 109–123.

[159] Kapočiūte-Dzikiene, J., Utka, A., and Šarkute, L. (2015). Authorship attribution of internet comments with thousand candidate authors. *Communications in Computer and Information Science*, *538*, 433–448. doi:10.1007/978-3-319-24770-0_37

[160] Nini, A. (2018). An authorship analysis of the Jack the Ripper letters. *Digital Scholarship in the Humanities*.

[161] Thisted, R., and Efron, B. (1987). Did Shakespeare write a newly-discovered poem? *Biometrika*, *74* (3), 445–455.

[162] Stańczyk, U., and Cyran, K. (2007). Machine learning approach to authorship attribution of literary texts. *International Journal Of Applied Mathematics And Informatics*, *1* (4), 151–158.

[163] Holmes, D. D. I. (1992). A stylometric analysis of Mormon scripture and related texts. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, *155* (1), 91–120.

[164] Burns, K. (2006). Bayesian inference in disputed authorship: A case study of cognitive errors and a new system for decision support. *Information Sciences*, *176* (11), 1570–1589.

[165] Stamatatos, E., Rangel, F., Tschuggnall, M., Stein, B., Kestemont, M., Rosso, P., and Potthast, M. (2017). Overview of PAN 2018 Author Identification, Author Profiling, and Author Obfuscation, *10456*, 267–285.

[166] Juola, P., and Stamatatos, E. (2013). Overview of the Author Identification Task at PAN 2013. In *CLEF (Working Notes)* 877–897.

[167] Mosteller, F., and Wallace, D. L. (2012). *Applied Bayesian and classical inference: the case of the Federalist papers*. Springer Science & Business Media.

[168] Koppel, M., Schler, J., and Argamon, S. (2008). Computational Methods in Authorship Attribution. *Journal of the American Society for Information Science and Technology*, *60* (1), 9–26.

[169] Yang, M., Chen, X., Tu, W., Lu, Z., Zhu, J., and Qu, Q. (2018). A Topic Drift Model for authorship attribution. *Neurocomputing*, *273*, 133–140.

[170] Koppel, M., Schler, J., Argamon, S., and Messeri, E. (2006). Authorship attribution with thousands of candidate authors. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* 659–660.

[171] Qian, T. Y., Liu, B., Li, Q., and Si, J. (2015). Review Authorship Attribution in a Similarity Space. *Journal of Computer Science and Technology*, *30* (1), 200–213.

[172] Savoy, J. (2012). Authorship Attribution Based on Specific Vocabulary. *ACM Transactions on Information Systems*, *30* (2), 1–30.

[173] Gressel, G., Hrudya, P., Surendran, K., Thara, S., Aravind, A., and Poornachandran, P. (2014). Ensemble Learning Approach for Author Profiling.

[174] Marquardt, J., Davalos, S., and Washington, T. (2014). Age and Gender Identification in Social Media. Retrieved from http://www.uni-weimar.de/medien/webis/research/events/pan-14/pan14-talks/pan14-author-profiling/marquardt14-poster.pdf

[175] Weren, E. R. D. D., Moreira, V. P., and de Oliveira, J. P. M. P. M. (2014). Exploring Information Retrieval Features for Author Profiling. In *CLEF (Working Notes)* Vol. 1180, 1164–1171.

[176] Tweedie, F. J., and Baayen, R. H. (1998). How Variable May a Constant be? Measures of Lexical Richness in Perspective. *Computers and the Humanities*, *32*, 323–352.

[177] Koizumi, R., and In'nami, Y. (2012). Effects of text length on lexical diversity measures: Using short texts with less than 200 tokens. *System*, *40* (4), 522–532.

[178] Stamatatos, E., Fakotakis, N., and Kokkinakis, G. (1999). Automatic Authorship Attribution. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics* 158–164. Stroudsburg, PA, USA: Association for Computational Linguistics.

[179] Hoover, D. L. (2003). Another perspective on vocabulary richness. *Computers and the Humanities*, *37* (2), 151–178.

[180] El-Kahlout, İ. D., and Oflazer, K. (2006). Initial explorations in English to Turkish statistical machine translation. *Proceedings of the Workshop on Statistical Machine Translation - StatMT '06*, (June), 7–14.

[181] Çakici, R. (2012). Morpheme Segmentation in the METU-Sabancı Turkish Treebank. *Proceedings of the 6th Linguistic Annotation Workshop*, (July), 144–148.

[182] Layton, R., Watters, P., and Dazeley, R. (2013). Local N-Grams for Author Identification. In *CLEF*.

[183] Escalante, H. J., Solorio, T., and Montes-y-Gomez, M. (2011). Local Histograms of Character N-grams for Authorship Attribution 288–298.

[184] Akimushkin, C., Amancio, D. R., and Oliveira, O. N. (2018). On the role of words in the network structure of texts: Application to authorship attribution. *Physica A: Statistical Mechanics and its Applications*, *495*, 49–58.

[185] Burrows, J. (2002). “Delta”: a Measure of Stylistic Difference and a Guide to Likely Authorship. *LLC*, *17*, 267–287.

[186] Hoover, D. L. (2004). Delta Prime? *Literary and Linguistic Computing*, *19* (4), 477–495.

[187] Eder, M. (2015). Taking stylometry to the limits: Benchmark study on 5,281 texts from" patrologia latina. In *Digital Humanitiess*.

[188] Zhao, Y., Zobel, J., and Vines, P. (2006). Using relative entropy for authorship attribution. *Airs, Lncs 4182*, *4182*, 92–105.

[189] Pavlyshenko, B. (2013). Classification analysis of authorship fiction texts in the space of semantic fields. *Journal of Quantitative Linguistics*, *20* (3), 218–226.

[190] Ng, A. Y., and Jordan, M. I. (2008). On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. *Neural Processing Letters*, *28* (3), 169–187.

[191] Loh, W.-Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1* (1), 14–23.

[192] Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1* (1), 81–106.

[193] Rokach, L., and Maimon, O. (2005). Top-Down Induction of Decision Trees Classifiers—A Survey. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, *35* (4), 476–487.

[194] Hastie, T., Tibshirani, R., and Friedman, J. (2009). Random Forests. In *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* 587–604. New York, NY: Springer New York.

[195] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2* (5), 359–366.

[196] Işık, Ş., Özkan, K., Gerek, Ö. N., and Doğan, M. (2016). A new subspace based solution to background modelling and change detection. *International Journal of Intelligent Systems and Applications in Engineering*, (December), 82–82.

[197] Cevikalp, H., Neamtu, M., Wilkes, M., and Barkana, A. (2005). Discriminative common vectors for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27* (1), 4–13. doi:10.1109/TPAMI.2005.9

[198] Hall, D., and Ramage, D. (2009). Scala Breeze Library.

[199] Argamon, S., and Juola, P. (2011). Overview of the International Authorship Identification Competition at PAN-2011. *CLEF*.

[200] Potthast, M., Braun, S., Buz, T., Duffhauss, F., Friedrich, F., Gülzow, J. M., Köhler, J., Lötzsch, W., Müller, F., Müller, M. E., Paßmann, R., Reinke, B., Rettenmeier, L., … Hagen, M. (2016). Who Wrote the Web? Revisiting Influential Author Identification Research Applicable to Information Retrieval. In N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, G. M. Di Nunzio, … G. Silvello (Eds.), *Advances in Information Retrieval. 38th European Conference on IR Research (ECIR 16)* Vol. 9626, 393–407. Berlin Heidelberg New York: Springer.

[201] Kourtis, I., and Stamatatos, E. (2011). Author identification using semi-supervised learning. *CLEF 2011: Proceedings of the 2011 Conference on Multilingual and Multimodal Information Access Evaluation (Lab and Workshop Notebook Papers), Amsterdam, The Netherlands*.

[202] Custódio, J. E., and Paraboni, I. (2018). EACH-USP Ensemble Cross-Domain Authorship Attribution. In *Working Notes Papers of the CLEF*.

[203] Murauer, B., Tschuggnall, M., and Specht, G. (2018). Dynamic Parameter Search for Cross-Domain Authorship Attribution. In *Notebook for PAN at CLEF* 1–6.

[204] Halvani, O., and Graner, L. (2018). Cross-Domain Authorship Attribution Based on Compression. In *Notebook for PAN at CLEF*.

RESUME

Hayri Volkan Agun

Yenibağlar Mh. Tutanlı Sk. 47/3
Tepebaşı, Eskişehir, Turkey
Email: volkanagun@gmail.com

Personal

Birthplace: Erzurum, Turkey, 1982

Research Interest

Research interests include text mining, machine learning, and natural language processing.

Education

PhD. Student in Computer Science Department of Illinois Institute of Technology, 2008-2009

Master in Computer Engineering, Trakya University, 2008. Thesis : A Machine Learning Application of Probabilistic Graphical Models in Natural Language Processing

Bachelor. in Computer Engineering, Anadolu University, Feb. 2005 Thesis: Fuzzy Logic, Neural Networks, Artificial Intelligence

Experience

Anadolu University - Software Developer - BAP Project, Sep. 2015 – Mar. 2016 Eskisehir, Turkey

Anadolu University - Software Developer - Student Information System, Dec. 2013 - Jan. 2015 Eskisehir, Turkey

TrakiaBilisim, Tekno Park, Campus of Aysekadin, Trakya Üniversitesi/Edirne, 2010-2012

Research Assistant, Trakya University, Computer Engineering Department., Edirne, Turkey, April 2006 - 2008

Publications

Agun, H. V., Yilmazel, S., and Yilmazel, O. (2017). Effects of language processing in Turkish authorship attribution. *2017 IEEE International Conference on Big Data (Big Data)*, 1876–1881.

Agun, H. V., and Yilmazel, O. (2017). Document embedding approach for efficient authorship attribution. In *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)* (pp. 194–198). IEEE.

Uzun, E., Agun, H. V., & Yerlikaya, T. (2013). A hybrid approach for extracting informative content from web pages. Information Processing and Management, 49(4).

Uzun, E., Güner, E. S., Kiliçaslan, Y., Yerlikaya, T., and Agun, H. V. (2014). An effective and efficient web content extractor for optimizing the crawling process. *Software - Practice and Experience*, *44*(10).

Agun H. V. and Y. Kılıçaslan (2011) Morphological Disambiguation via Conditional Random Fields, *Proceedings of 2nd International Symposium on Computing and Engineering (ISCSE)*.

Kılıçaslan, Y., E. Uzun, H. V. Agun and E. Uçar, (2007) Automatic Acquisition of Subcategorization Frames for Turkish with Purely Statistical Methods, *Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications (INISTA*), 11-15.

Arda D., Agun V., Demiralp M. (2006) Effect of the Differentiation on the Additivity of High Dimensional Model Representation, *International Conference on Numerical Analysis and Applied Mathematics (ICNAAM)*, 36-39.

Agun V., Arda D., Demiralp M., High Dimensional Model Representation Based Data Partitioning Implementations Via Parallel Computations, *International Conference on Numerical Analysis and Applied Mathematics (ICNAAM)*, 20-23.