ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCES

Authorship Attribution Model for Amharic Documents using Machine Learning

Baher Hussen Geletu

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

November, 2020

Addis Ababa University

College of Natural Sciences

Baher Hussen Geletu

Advisor: Dida Midekso (PhD)

This is to certify that the thesis prepared by *Baher Hussen*, titled: *Authorship Attribution Model for Amharic Documents using Machine Learning* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	Name	Signature	Date
Advisor:	_____	_____	_____
Examiner:	_____	_____	_____
Examiner:	_____	_____	_____

Abstract

These days, text documents are being produced anonymously through different sources like the Internet. They are available in different forms without the rightful owner of the text being known. These anonymous texts can be emails, letters, harassing messages, suicide notes or literary works created using different languages. Identifying the true author of such an anonymous text involves analyzing the writings through the various authorship attribution techniques. However, the forms of these texts, and the type and nature of the language that are used to create them make the process challenging. So far, in the Amharic language, researchers tackled the problem of topic based text classification to some extent. However, style based text classification tasks, like the authorship attribution problem, hasn't been given much consideration.

This study is aimed at designing an Amharic authorship attribution model that is capable of identifying authors of anonymous Amharic documents using machine learning. The architecture is composed of two phases (a training and an attribution phase) comprising different components: Preprocessing, Feature Extraction, Feature Concatenation, Dimension Reduction, Classifier Training, Author Profiling and Authorship Attribution. Different types of n-gram features (word, character, part of speech, punctuation and space), the different combination of these n-grams and n-gram based poem specific features are considered. The training phase involves extracting sets of features from the author dataset creating an author profile to train a classifier with. The attribution phase involves extracting sets of features from a given anonymous test document and attribute an author from a set of candidate authors.

A prototype of the attribution model is developed to test and evaluate its practicality. The model is experimented using a dataset of more than 2,000 documents of 20 different authors and 120+ poems of 2 poets for the different n-gram features. The model is tested using support vector machine and has achieved an accuracy of 86.77 % for the combination of char 3-gram and word_plus_pos 4-gram features using support vector machine classifier for dataset 1 and an accuracy of 0.96 for poem dataset for poem specific features.

Keywords*: Machine Learning, Text Classification, Stylometry, Authorship Analysis, Authorship Attribution*

ACKNOWLEDGEMENT

I would first like to thank my thesis advisor Dida Midekso (PhD) of the department of computer science at Addis Ababa University. The door to Dr. Dida office was always open whenever I ran into a trouble spot or had a question about my research. Starting from reshaping this thesis, his support and encouragements were high and he consistently allowed this paper to be my own work, but steered me in the right direction whenever he thought I needed it.

I would also like to thank my friends for the support they provided me with in sharing their knowledge and experience, and their positive comments and suggestions which was a substantial help in bringing this thesis to an end.

Finally, I must express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Abbreviations and Acronyms

AA	Authorship Attribution
NLTK	Natural Language Toolkit
ML	Machine Learning
TC	Text Classification
POS	Part of Speech
TF	Term Frequency
TF/IDF	Term Frequency/Inverse Document Frequency
SPCA	Sparse Principal Component Analysis
SVM	Support Vector Machine
NB	Naïve Bayes
SVC	Support Vector Classifier

Chapter One: Introduction

1.1 Background

In recent years, there has been developing consideration to the different applications related with machine learning techniques [1]. These techniques are concerned with automatic discovery of regularities or patterns in data through the use of computer algorithms and with the use of those patterns to take actions such as classifying data into different categories. An interesting application of these methods is author attribution and its essential thought is to allot/ attribute a composed piece of text to one author out of a possible group of authors. It is a long explored range of research going back to the early 1960s with a celebrated case, the Federalist Papers case [2].

More imperatively, the plethora of accessible electronic writings uncovered the potential of authorship analysis in different applications in diverse areas [3], including intelligence (e.g., attribution of messages or proclamations to known terrorists, criminal law (e.g., identifying writers of harassing messages, linking different messages by authorship, verifying the authenticity of suicide notes) and civil law (e.g., copyright disputes, and computer forensics (e.g., identifying the authors of source code of malicious software) in addition to the traditional application to literary research (e.g., attributing anonymous or disputed literary works to known authors).

Authorship analysis deals with the individual style of authors and incorporates three major areas [4]: *Author identification/Attribution, Author profiling and Author clustering*. *Author identification* involves, given a set of candidate authors for whom a few writings of undisputed authorship exist, attribute writings of unknown authorship to one of the candidates. This could be connected primarily to forensic applications and literary analysis. *Author profiling is* the extraction of demographic data such as gender, age, etc. about the authors. *Author clustering* is the segmentation of writings into stylistically homogeneous parts which is mainly used to distinguish different authors in collaborative writing and detect plagiarism without a reference corpus (intrinsic plagiarism detection). Author clustering can also be used to detect changes in the personal style of a certain author by chronologically examining their works.

Published authors regularly have a particular way of writing that's apparent to human readers [5]. Given a little body of text, the capacity to precisely distinguish authors can be important in research and other areas where devouring and synthesizing huge sums of relevant information is vital, since an individual interested in a little body of text by an author is likely to be fascinated by larger, related works composed by the same author. Researchers assume that all authors have specific style characteristics that are outside their conscious control [6]. Thus, on the premise of those linguistic patterns and markers, the author of a document can be identified.

Despite that numerous computational linguistic tasks can be solved precisely only relying on endeavors of domain-experts, it is expensive, time consuming, and possibly the most restrictive way for authorship attribution (AA), additionally, it provides no explicit measure how attributions are made [7]. The alternative way could be a manually composed set of rules able to require attribution decisions automatically. Unfortunately, rule-based systems as a rule are exceptionally complex, clumsy, and hence not robust to any changes in the domain, language or author characteristics, therefore it is or maybe troublesome to form any updates. Moreover, when dealing with hundreds or thousands of candidate authors, the possibility to create an effective rule set goes far beyond human potential limits. Ultimately, authorship attribution tasks can be solved using the machine learning: i.e. by training the classifiers and later using them to predict the authorship of unseen texts. Besides, it can be effectively balanced to modern applications or domains and even generalized well to drifts within the author characteristics.

1.2 Motivation

Identifying the author of a text is a genuine, and recurring issue that reappears again and again over numerous distinctive fields. Archeologists and historians regularly recover and attempt to distinguish unattributed writings [8]. Sometimes, one has a document and needs to know not what it's about, but who wrote it -- for example, a teacher looking at a possibly plagiarized paper, or a policeman looking at a ransom note [9].

The authorship attribution method can be valuable in applied ranges such as law and news coverage where the recognizable proof of the genuine author of a piece of text (such as a ransom note) may be able to save lives or help accuse wrongdoers [10]. Given its application and the fact that there have been made no effort in the authorship attribution task for the Amharic language motivated us to propose this work.

1.3 Statement of the Problem

These days, there's a rapid development of text in electronic form in poems, articles, news, etc. [11]. Most of this content is provided anonymously or under unverified names. In the framework of forensic applications, it is required to gather texts written by the same author or track texts written under diverse names but belonging to the same individual. In addition, there are various copyright dispute cases where numerous individuals claim the authorship of texts. Authorship identification backed by computational analysis of texts draws in growing consideration since it may offer quick answers to these problems.

In the easiest form of the problem, we are given cases of the writing of a number of candidate authors and are inquired to decide which of them wrote a given anonymous text [12]. In this straightforward form, the authorship attribution problem fits the standard modern paradigm of a text categorization problem. In any case, real-life authorship attribution problems are rarely as exquisite as direct text categorization problems, in which we have a small closed set of candidate authors and basically unlimited training text for each.

AA could be a kind of classification problem but it is different from text classification (TC), because in AA the writing style is also vital besides the content of the text, which is the only factor used in text classification [13]. In addition, the features in AA task are not deterministic as in text

classification. Also, with different data such as books and articles, the classifiers and feature sets may behave differently. Therefore, these differences make AA task more challenging. In text classification task, the texts should be assigned to one or more predefined classes based on the topics, while in AA task, the texts should be assigned to one or more predefined classes based on the authors. So far, several studies addressed the task of AA through numerous classification techniques and algorithms in different languages. These techniques provide the platform to make use of the different types of stylometric features and perform accurate identification of authors of an anonymous texts. Support vector machines, Neural Networks, Burrow's method and Naïve Bayes are among these AA techniques [14].

On the other hand, each language possess its own distinctive way of writing style which is also unique for each author. In the Amharic language, writing style differs from several other languages in many different ways. Starting from the character set used in the writing system of the language to the structure of sentences, from the set of punctuations used which are unique to the language to the morphological complexity and richness, unique grammatical rules and many more characteristics of the language allows authors who write using the language produce a huge quantity of styles which are unique not only to the author but also to the language itself. As different author means different style, different language means different style. As David Recine [15] puts it, "*The truth is that every writing style in every language is logical—and each one has its own logic*.", meaning that when one writes using a certain language he/she should follow the grammatical and other sets of rules which are specific to the language.

In addition to the challenges that an AA problem could inflict, Amharic is a language for which very few computational linguistic tools or corpora (such as lexica, part-of-speech taggers, parsers or tree-banks) exist [16]. This problem, which is shared by a number of the so called "under-resourced languages" has proven to be a bottleneck when it comes to promoting the use of computers and the Internet in the language. It is always a troublesome assignment to develop modern linguistic resources without access to already existing ones.

The authorship attribution problem is applicable in many different aspects providing answers to questions of ownership over anonymous texts in different forms. Performing authorship attribution studies across different language reveals different characteristics of style features used to represent an author's writing in each type of language. Moreover, such studies that focus on style features

flags the means for more studies on style based text classification tasks. So far, research works on the authorship attribution problem consider mainly the English language and some other languages like the Arabic and Punjabi. However, authorship analysis problems which classify Amharic texts based on author related aspects have not been studied. Thus, in this work, we proposed to tackle the problem of authorship attribution in the language using machine learning techniques.

1.4 Objectives

General Objective

The general objective of this research is to develop a model for authorship attribution of Amharic documents using machine learning.

Specific Objectives

The specific objectives that we will consider to meet the general objective are to:

- Evaluate the different techniques and approaches employed so far in authorship attribution,
- Study the different features that can be used to characterize an author in the Amharic language,
- Prepare a set of training documents,
- Develop or adapt feature selection method for extracting a set of features,
- Perform a comparison of currently available author attribution methods,
- Develop a classifier for the classification model to classify documents based on author,
- Develop a prototype of the model,
- Test and evaluate the model through the prototype.

1.5 Methods

With the aim of achieving the research objectives outlined above, the following methodologies will be applied.

Literature Review

Literature review or state-of-the-art analyses with respect to the Amharic language, text authorship attribution and exploring the different techniques of feature selection methods and classification algorithms are done.

Data collection

Data collection will be a critical task of the research since obtaining the required data will be vital in the development, training and testing of the system. Data is collected from multiple sources to create two datasets. To build the first dataset articles from kumneger… magazine and the reporter newspaper are collected. The second dataset is created using collection of poems written by two poets.

Testing and Evaluation

The developed prototype is tested using set of collected documents of known Amharic authors. The model performance is evaluated using the evaluation metrics: precision, recall and F-measure.

1.6 Scope and Limitations

The scope of this work is to develop a model of an authorship attribution system for Amharic documents. This documents include news, articles and poems. This research work considers only closed set attribution of Amharic documents, i.e., open set attribution of documents is not included in this work.

1.7 Application of Results

The importance of authorship identification lies on its wide applications [17]:

- Finding the original author of widely reprinted articles, as well as plagiarized paragraphs.

- Providing a new way to recommend to a reader the authors who have a high similarity of writing style with his/her favorite authors or anonymous writers,
- Resolving historical questions of unclear or disputed authorship, and
- Deducing the writer of inappropriate communications that were sent anonymously or under a pseudonym (e.g. threatening or harassing e-mails).

1.8 Organization of the Thesis

The rest of this thesis is organized as follows. In Chapter Two, review of literatures and methodologies related to authorship attribution is presented. Chapter Three presents related research works which are in the area of the problem of authorship attribution. In Chapter Four, the design of the authorship attribution model with the discussion of each of the components is presented. Experimentation together with evaluation results of the system is presented in Chapter Five. In Chapter Six, conclusion, contribution of this study and future works are presented.

Chapter Two: Literature Review

2.1 Introduction

In this chapter, we present essential review and analysis of literatures and methodologies which are relevant to the authorship attribution problem. Reviews of natural language processing, the Amharic language, machine learning, text classification, authorship analysis, authorship attribution, feature extraction and selection in AA and authorship attribution approaches are included.

2.2 Natural Language Processing

Natural language processing (NLP) is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications. As it is described in [18], in the linguistic analysis of a digital natural language content, it is essential to clearly define the characters, words, and sentences in any document. Defining these units presents diverse challenges depending on the language being processed and the source of the documents, and the task isn't trivial, particularly when considering the variety of human languages and writing systems. Adding more to this concern, Patrick J. [19] asserts, the "human language can be a very difficult system to study, because it combines a maddening degree of variability with surprisingly subtle regularities. In order to analyze language with computers, it is usually necessary to make simplified models of language for analysis".

These days' lots of poems, articles, news, etc., are being made namelessly, different individuals are in a dispute over a single piece of written text claiming its authorship, threatening or harassing e-mails are being sent namelessly and historical questions of hazy or disputed authorship still exist. Fortunately, this problem can be solved by the authorship attribution method which may be a long-standing and well-studied problem within the text classification task of NLP where the objective is to classify documents according to their authorship.

2.3 Poems and Articles

Poems

Text documents are produced in various forms each for different purposes as they are chosen by the specific author. Authors choose the different forms depending on the type of message they want their text to transmit or the type of audience they intend their message for. Many of these text can take one of the various forms, such as: emails, short messages like the SMS, articles, literatures, poems etc. While writing these texts authors produce a unique way of writing style for each form of texts intentionally or subconsciously. One of these forms of texts is the poem, which is artistic by its nature and used by many authors (poets) for conveying emotions, imaginations, history, experiences and ideas often written using rhymes and meters.

Poems differ from prose and other forms of texts in that they are written following a certain metrical structure [20]. These metrical structure involves rhymes and meters which are sets of rules that govern the arrangements and numbers of syllables in each verse (line) [21]. The main elements of a poem are verses, rhymes, meters and stanzas. Verses represent a single line in a poem, rhymes are repetition of sounds in the final stressed syllable of verses, meters are the basic structure of the verses and stanzas represent a group of verses usually separated by an empty space between other stanzas. Based on the number of verses stanzas can be identified into couplets (2 verses), tercets (3), quatrains (4), cinquain (5) etc.

Articles

Articles are types of writing which are, not very long or not very short, commonly as short as one or two paragraphs or as many pages long as the particular author wants it to be. Articles can be used to address different aspects since they are capable of conveying different opinions of writers, researches, news and reviews. They usually are designated by authors to carry information's like recent news of broad interest and are intended for the general audience. They are used to reflect a lot of topics and are produced in different formats. Given their briefness [22], articles appear in many different forms of texts like: Book Reviews, Literature Reviews, Issues, Journals, Magazines, Newsletters and Newspapers etc.

2.4 The Amharic Language

Amharic is the working language of the federal government of Ethiopia and it is broadly spoken throughout the nation as a first and a second language. It is the official language of the Amhara regional government. It is highly inflectional and quite dialectally diversified [23]. With millions of speakers, it is the second most spoken Semitic language within the World (after Arabic) and nowadays likely one of the five biggest on the African landmass (yet troublesome to decide, given the sensational population estimate changes in numerous African nations in recent years).

Amharic and Tigrinya languages draw common roots to the ecclesiastic Ge'ez still utilized by the Ethiopian Orthodox Church [24]. Both languages are written using the Ge'ez script, horizontally and left-to-right (in contrast to numerous other Semitic languages). Written Ge'ez can be followed back to at least the 4th century A.D. The primary versions of the script included consonants only, whereas the characters in later versions represent consonant-vowel phoneme pairs. In written Amharic, each syllable pattern comes in seven different forms (called orders), reproducing the seven vowel sounds. The first order is the basic form; the other orders are derived from it by more or less regular modifications indicating the different vowels. In the language, there are 33 fundamental forms giving 7*33 syllable patterns.

There's no agreed-upon spelling standard for compounds and the writing system uses multitudes of ways to indicate compound words [23]. Moreover, not all the letters of the Amharic script are entirely essential for the pronunciation patterns of the language; a few were basically inherited from Ge'ez without having any semantic or phonetic distinction in modern Amharic (although they do have in Ge'ez and Tigrinya). There are numerous cases where various symbols are used to indicate a single phoneme, as well as words that have amazingly diverse orthographic form and somewhat distinct phonetics, but the same meaning. For example, most of the labialised consonants are basically repetitive, and actually there are only 39 context independent phonemes by removing the redundant ones, out of the 275 symbols of the script only 233 remain.

The script which Amharic uniquely uses (provided under the annex section), "fidel", is conveniently written in a tabular format of seven columns [25]. In the first column the basic form is represented, and the other orders are derived from the basic form by regular modifications designating the different vowels.

The Amharic number system consists of twenty characters. These characters represent numbers one to ten, multiples of ten (twenty to ninety), hundred and thousand. The numbering system is not suitable for arithmetic computation [26] because there is no comma, decimal point or representation for zero (0). Amharic numbering system is used in dates usually in calendars, else western numerals are used in most literature these days [27].

Around 10 punctuation marks exist in the Amharic writing system [28]. However, only few of them are practically used, particularly in computer systems [29]. The basic punctuation marks in Amharic writing system are

- Hulet Neteb (two square dots arranged like colon, **:**, used as a word-separator) and
- Arat Netib (sentence-separator, four square dots arranged in a square pattern **::)**.

The other common type of punctuation marks in the language are:

- 'Netela Sereze' (፣), an equivalent of comma, tailed by ASCII space separates lists in Amharic texts and
- 'Derib Sereze' (፤), which is the equivalent of semi-colon, can also be used as a list separator.

As it is indicated in [27], the Amharic writing system has borrowed a few punctuation marks from foreign languages like the question mark '?'. The full list of punctuation marks in the language are provided under the Annex section.

There are a number of issues related with the Amharic writing system making the natural language processing of Amharic documents a bit more challenging [26]. Some of these problems are:

- *Redundancy of some characters*.
- *Compound words*: there is no standard way of writing Amharic compound words [27].
- S*pelling variation* of same words (the same word can be written in various forms) [29] is another issue.
- Inconsistency of abbreviations of Amharic words [30].

All these mentioned issues pose challenges since the same word is treated in a completely different form within the process of feature preparation creating style differences between authors for a classifier.

Amharic Morphology

Amharic is one of the most morphologically complex languages which exhibits a root-pattern morphological phenomenon. A root is a set of consonants comprising a basic lexical meaning. A stem is formed by inserting patterns consisting of a set of vowels among the consonants of a root. Adding to this morphological features, in the Amharic script different affixes (prefix, infix, suffix and circumfix) are used creating inflectional and derivational morphemes [25]. The morphological complexity of the language is way better caught on by looking at the word-formation process through inflection and derivation.

Inflection

Inflection is the process of word-formation by modifying words to express different grammatical categories. Nouns, adjectives, pronouns and prepositions can be inflected. For example, Amharic nouns can be inflected for number, definiteness, cases (accusative/objective, possessive/genitive) and gender [25]. Amharic adjectives, as that of nouns, can be inflected for number, definiteness, cases and gender. On the contrary, verbs in Amharic can be inflected for any combinations of person, gender, number, case, tense/aspect and mood. As a result, from a single verbal root, thousands of verbs (in surface forms) will be generated. Verbal stems in Amharic consists a "root + vowels + template" merger [25]. For example, the root verb "ሰብር" (sbr) + ee + CVCVC forms the stem seber ('broke'). [31] shows from the root word "ሰብር" (sbr/to break), it is possible to derive verbal stems like "ሰብር" (säbr), "ሰበር" (säbär), "ሰባብር" (säbabr) and ተሰባብር (täsäbabr), etc. and words like "ሰበረው" (säbäräw), "ሰበረች" (säbäräč), "ሰበርን" (säbärn), "አሰበረ" (assäbärä), "ተሰበረ" (täsäbärä), "አልሰበረም" (alsäbäräm), "ካልተሰበረ" (kaltäsäbärä), "የሚሰበር" (yämisäbär), etc.

Derivation

Through the process of derivation, Amharic nouns can be derived from adjectives, verbal roots by embedding vowels between consonants, stems, stem-like verbs and nouns themselves [25]. Few primary adjectives exist in the language. However, from nouns, stems, compound words and verbal roots adjectives can be derived. Adjectives can moreover be derived either from roots by intercalation of vocalic components or attaching a postfix to bound stems. Amharic verbs can also be derived from different verbal stems.

The development of Amharic language processing resources and digital information access and storage facilities is getting an improved attention due to the need for accurate and fast information access [23]. So far, some work on Amharic has been done in areas such as:

- Word formation [32], stemming [33], Treebank building [34],
- Character recognition [35] and text classification based on topics [36].

Stemming/ Lemmatization

Stemming is understood to be an important preprocessing task usually followed by text classification works to improve classification accuracy to a significant point. Stemming is more central to languages like the Amharic which possess rich and inflectional morphology. Stemming can be defined as the process of combining multiple word variants into semantically similar roots by removing derivational and inflectional affixes. For example, the Amharic word "የቤታችን" will be stemmed to the root word "ቤት". Stemmers for the Amharic language has been developed using different methods and algorithms. In the language, stemmers have been used in studies like information retrieval [37] and text classification [26] and has achieved significant outcomes. Lemmatization is a very similar task to stemming which involves the use of vocabulary and morphological analysis of words to get rid of inflectional endings and retrieve a dictionary form of the word (lemma). Lemmas are the base forms of all inflectional forms of words whereas a stem basically isn't. In Amharic, these two tasks are usually used interchangeably with one serving as a replacement for the other.

Syntactic Structure

A syntax is a set of rules to be followed in ordering words to create sentences which are both logical and meaningful. In natural languages, syntax represents how words and morphemes are grammatically arranged to form sentences. Syntactic structures are different from one language to another. In the Amharic language, the standard word order is subject-object-verb. Yet, in some cases, if the object is topicalized, it may precede the subject and the order will become object-subject-verb [38]. Different studies categorize Amharic words into different categories of word classes. [39] categorizes Amharic words into five broad categories: Nominals (noun-like words), Verbal's (verb-like words), words sharing morphological features of Nominals and Verbal's (verbo-nominals), Positional complexes (nominal, verbo-nominal or nomino-verbal) and other

word classes. According to [40], Demeke categorizes Amharic words into four basic classes: nominals, verbs, adpositions-conjunctions and interjections which are further divided into subclasses and other studies like Baye (1987) and Mersiea Hazen (1935) divide Amharic word classes into different categories. On the other hand, part of speech tagger studies [41] in the language define eleven basic classes or tagsets (nouns, pronouns, adjectives, adverbs, verbs, prepositions, conjunctions, punctuation, numeral, unclassified and interjections) to tag words. In the language several works like parse trees, part of speech taggers and context free grammars are developed for syntactic processing.

Semantic Structure

In a language, semantics is concerned with meaning at the levels of words, phrases and sentences. [42]. Lexical and compositional semantics are the two types of semantics in natural languages. Lexical semantics deals with the meanings of words and other lexical expressions. Compositional semantics is concerned with how lexical meanings combine to generate phrasal meaning. In the Amharic language several studies like [31] tried to address the semantics of the language though the construction of semantic networks. A different study [43] applied the concept of semantics to indexing Amharic texts and [44] performed semantic analysis for text summarization task.

Poems in Amharic

In the Amharic language poems are considered significant part of literature. Poets use poem to deliver various message starting from the day to day life of the community to the social aspects, cultural practices and political concerns. Poetry in Amharic is more than just a writing with a unique set of rhyming technique. Several linguists of the language tried to identify poems into different classes and define sets of rules to write an Amharic poem [45]. Linguists like Berhanu Gebeyehu, Mersiea Hazen, Alemayehu Moges and Mengistu Lemma are some of the various linguists who wrote about the basic constituents and structure of a poem [46]. Amharic poems components include ስንኝ (verse), ቤት (rhyme), አረፍተ ስንኝ (stanza), ቀለም (syllable) and ምጣኔ (meter). In the language poems are grouped into different classes (such as: የወል ግጥም፣ ሠደፌ ግጥም፣ የጀግና ግጥም etc.) depending on the syllable count in each verse. Different linguists define different sets of classes to categorize the poems based on their own sense of understanding leading to the creation of a number of class names and inconsistencies in labeling a given poem to a given class [47]. These and many more disputes between various linguistics on the understanding of the basic

notion to writing poems created a huge gap in setting the rules a given poet has to follow in writing poems. Starting from the conception of segmenting phrases in poems to verses and syllable counts in verses are the main challenges in the recognition of poems. Furthermore, the concept of syllables in poems is not clearly defined with each linguist defining their own way of counting syllables in a verse, with [46] concluding that the understanding of syllables in poems depend on letters (fidels) than phonemes as it is in linguistics and needs more studies that could combine the concept of syllables in both linguistics and poetry.

2.5 Text Classification/Categorization

Text classification is the task of classifying a document under a predefined category. Text classification (TC) has been broadly studied in different fields such as machine learning, data mining and information retrieval, and is used in vast number of applications in various domains such as medical diagnosis, document organization, image processing, etc. [48]. Putting this formally, if d_i is a document of the entire set of documents D and $\{c_1, c_2, c_3 ... c_n\}$ is the set of all the categories, then text classification will assign one category cj to a document di.

In text classification, documents depending upon their characteristics can be labeled for one class or for more than one class. During classification, if a document is assigned to only one class from a set of classes, it is called “single-label” and if the document is assigned to more than one class, it is called “multi-label”. A number of approaches to text categorization has been proposed [49]. The objective of text categorization methods is to relate one or more of a given set of categories to a specific document. These methods differ on how they represent documents and on how they decide which category to assign to a particular document.

The significance of text categorization increased within the late ‘80s and early ‘90s with the need to organize the progressively bigger amounts of digital content being dealt with in organizations at all levels [49]. Since then, newswire filtering, patent classification, and web page classification are frequently pursued applications of the text classification technology. All these applications involve a definite thematic content, meaning that classes tend to accord with topics. However, text classification technology has also been applied to domains that are not thematic in nature, among which are:

- Spam filtering (i.e., the grouping of personal e-mail messages into LEGITIMATE and SPAM) [50];
- Authorship attribution (i.e., the automatic identification of the author of a text among a predefined set of candidate authors) [51];
- Author gender detection (i.e., deciding whether the author of the text is a male or a female) [52];
- Genre classification (i.e., identifying of non-topical nature of a text, such as determining if a product description is a product review or an advertisement;) [53];
- Survey coding (i.e., classification of respondents to a survey based on textual answers they have returned to an open-ended question) [54]; or
- Affective rating (i.e., deciding if a product review is thumbs up or a thumbs down) [55].

2.6 Machine Learning

Up until the late 1980s the foremost prevalent approach to text classification, at least within the community of real-world applications, was a knowledge engineering one, comprising in manually characterizing a set of rules encoding expert knowledge on how to classify documents under the given categories [56]. However, in the 1990s, this approach has progressively lost its popularity in the research community to the machine learning (ML) paradigm.

Machine learning, in its basic essence, is a paradigm that refers to learning from past experience (previous data) to increase future performance with its sole focus of the field being automatic learning methods [57]. Learning refers to the automatic adjustment or enhancement of an algorithm based on past "experiences" without any external help from a human. It is an interdisciplinary field building upon many ideas from many different fields such as cognitive science, information theory, optimal control, artificial intelligence, statistics, optimization theory, and other disciplines of science, mathematics and engineering [58].

Learning in the ML paradigm can be achieved through using previously labeled training data by domain experts (supervised learning), by using unlabeled training data (unsupervised learning), a combination of both (semi-supervised learning), and through reinforcement learning, which is a rewarding system that strengthens strategies for making good decisions and deteriorates inferior decisions [59].

- **Supervised learning:** Input data (training data) has a pre-determined label (e.g. Positive/Negative, Spam/Not Spam and True/False etc.). A classifier is built and trained to predict the class/label of a given test data. The classifier's parameter values are tuned (adjusted) to achieve the appropriate level of accuracy.
- **Unsupervised learning:** there is no labelled input data (training data). In unsupervised learning, a classifier is designed by presuming existing patterns in the training data.
- **Semi-supervised learning:** the training dataset contains both labeled and unlabeled data. The classifiers here train and learn the patterns to classify and label the data, and also make predictions.
- **Reinforcement learning:** The algorithm is trained to map action to situation so that the reward or feedback signal is maximized. The classifier is trained to find most rewarding actions through trial and error, and choose.

Table 2.1 shows the comparison of machine learning technologies

Table 2.1: Comparison of Machine Learning Techniques

Learning Method	Preprocessing	Distinction Norm	Learning Algorithm
Supervised	- Classification - Regression - Estimation	Computational Classifiers	Support Vector Machines
		Statistical Classifiers	Naïve Bayes Bayesian networks
		Connectionist Classifiers	Neural Networks
Unsupervised	- Clustering - Prediction	Parametric	K-Means
		Non-Parametric	X-means
Reinforcement Learning	- Decision Making	Model Free	Q-learning R-learning
		Model Based	TD-Learning Sarsa Learning

The machine learning technique has previously been applied to the Amharic language by Lars Asker et al., [36] to classify Amharic texts and this same technique has also been applied on some other works like [60] for the language.

The use of machine learning classifiers marked a vital turning point in authorship attribution studies. The application of authorship attribution methods is straightforward [61]: training texts are represented as labeled numerical vectors and learning methods are used to find boundaries between classes (authors) that minimize some classification loss function.

2.7 Authorship Analysis

Authorship analysis is the process of examining the characteristics of a piece of work in order to draw conclusions on its authorship. In linguistics, the authorship analysis study roots from stylometry, which refers to statistical analysis of literally style [62]. It is also defined in [63] as, the statistical study of linguistic and computational characteristics of the written documents of individuals.

Authorship analysis measures some textual features and avoids distinguishing between texts written by different authors. Authorship analysis studies can be classified into three categories as authorship attribution/identification, authorship profiling/characterization and authorship verification or similarity detection.

- **Authorship attribution**: determines the likelihood of a particular author having written a piece of text by analyzing other texts written by that same author [1].
- **Authorship profiling**: determines the characteristics of the author (the author's profile) that produced a given piece of text. These characteristics include gender, educational background, cultural background and language familiarity [62].
- **Similarity detection**: compares multiple pieces of work and determines if they are produced by a single author or not without the need to identify the author. Similarity detection is regularly utilized within the context of copyright infringement detection [62].

2.8 Authorship Attribution

Determining the author of a specific piece of text has raised methodological questions for centuries. The problem of authorship attribution, a problem with a long history and a wide range of application, is not only limited to scholars of humanities. But, it is also an interest to politicians, journalists, and lawyers [19]. The knowledge behind authorship attribution supported by statistics

is that by measuring some literary features, it is possible to distinguish between texts written by diverse authors [3]. According to [3], the first attempts to quantify the writing style go back to the 19th century, with the groundbreaking study on the plays of Shakespeare by Mendenhall [63], followed by statistical studies in the first half of the 20th century. Later, the study [2] on the authorship of "The Federalist Papers" was undoubtedly the most influential work in authorship attribution.

Authorship attribution can be defined as the task of inferring characteristics of a document's author from the textual characteristics of the document itself. It is an endeavor to induce the characteristics of the author of a piece of linguistic data. Authorship attribution is the procedure of deciding the author of a text when it is uncertain who wrote it [64]. It is of use when there is an argument between two or more individuals over who wrote something or in some cases where no one is able to assert who wrote the document.

In the literature, authorship identification is considered as a text categorization or text classification problem. Unlike typical document classification, however, in authorship attribution one does not desire to classify documents based on document content [65]. Instead, one desires to perform classification based upon author "style". While there are several case specific issues (such as genre, subject, register, time period, sample length, etc.) that must be considered when testing for authorship, the two most important factors from a machine learning perspective involve the choice of features and the selection of an appropriate classification method.

The AA task is divided into two different subtasks [66]. First task is identifying the most discriminative features to distinguish the writing styles of authors. The second task is finding the proper classification algorithm to detect the most probable author of a test document. The process starts by cleaning data followed by feature extraction and normalization [62]. Each suspected document is converted into a feature vector, the suspect representing the class label. Stylometric features are used to calculate values for features. Features extracted are classified into two groups: training sets and testing sets. The training set is used to develop a classification model whereas the testing set, assuming that the class labels are not known, is used to validate the developed model. Classifiers commonly used for classification in most AA studies include decision trees, neural networks and support vector machine.

AA studies are differentiated through the stylometric features used and the type of classifiers employed [62]. Numerous approaches have been proposed to undertake the problem of authorship attribution [67]. Traditional techniques involve using text analytics and natural language processing concepts to characterize an authors' writing style [4]. In recent years, more profound paradigms have been employed to handle this problem. Semantic and syntactic features are some examples of features not depending only on basic statistical analyses [7]. However, even though they're effective in specific contexts, deeper paradigms require a more complex data handling and a thorough effort which will not yield good results in generic scenarios [68].

2.8.1 Text preprocessing in Authorship Attribution

Text preprocessing, the task of transforming a raw text file, which is basically a sequence of digital bits, into a sequence of linguistically significant units: at the lowest level, characters representing individual graphemes in a written language, words comprising one or more characters, and sentences comprising one or more words [18]. Text preprocessing is an indispensable task in any NLP system, for the characters, words, and sentences at this phase are the fundamental units being passed to further processing stages.

Text preprocessing is a very important and tedious step in authorship attribution. This preprocessing step is vital in determining the quality of the feature extraction and classification stages [64]. In the process of categorizing texts by style, there are a number of text preprocessing tasks involved. However, in the literature, different text preprocessing stages are followed depending on the language of the text and the AA methods and approaches used. The following are the preprocessing tasks discussed:

- **Tokenization:** Tokenization is the process of splitting a stream of input text into meaningful elements. These elements are called tokens like symbols, words, phrases and so on. These extracted group of tokens act as an input for further processing like parsing and text mining. The task of tokenization is to convert a given text into tokens or morphemes by splitting the input text into pieces. The input texts are tokenized into a sequence of tokens by locating word boundaries from texts. In several NLP works, white space and punctuation marks are used as boundary markers. In authorship attribution studies punctuation marks are considered to be effective markers of an author's writing

style. The pattern and choice of punctuation marks an author prefers to use in writing a specific text is well thought out to represent the style of writing for that particular author. The Amharic language has its own punctuation marks like: arat netib (::), sost netib (…), derib serez (፧) and hulet netib (:) etc. When it comes to encodings, the fact that there exist multiple encodings for the same character sets in a particular language is much of a problem for tokenization. Therefore, to handle this character set dependency issue, it is required to convert the encodings of all texts in our dataset to the utf-8 encoding.

- **Stemming:** Stemming is the process of reducing an inflected word to its root or base form called stem. The stem may not be same as the morphological root of that word. Even if a stem isn't a convincing root, it is sufficient that related words can map to the same stem. A stemmer (stemming algorithm) is a program used to perform stemming.
- **Lemmatization:** Lemmatization is the task which involves morphological analysis of words, i.e. grouping together the various inflected forms of a word so they can be analyzed as a single item. In other words lemmatization methods try to map verb forms to infinite tense and nouns to a single form.
- **Filtering:** Filtering is commonly done on documents to remove some of the words which are not necessarily needed in a given study. A common filtering task is stop-words removal. Stop words are words (such as prepositions, conjunctions, etc.) that frequently appear in a text without having much content information.
- **Sentence Segmentation:** A sentence segmentation process, also known as sentence tokenization, which involves finding sentence boundaries between words in large texts is needed to segment sentences into sections so that the length of each sentence will be computed. Since sentence boundaries of written languages are punctuation marks, the process can be understood as dividing sentences through punctuations. It is clear that text segmentation issues are considered language-specific since many languages of the world exhibit different writing systems resulting in language specific and orthography specific features.
- **Text Distortion:** is the process of creating a new version of a text by masking a certain content of the original text. Text distortion is not the most common type of preprocessing task, but it has been considered in some text classification [67] tasks to mask words or characters that are not necessary for that specific study. Text distortion is the technique of

distorting or converting a given input text to a more suitable or distorted form needed by masking some characters, words, digits or symbols in the text. This technique is usually employed in topic based text classification works which can be used to mask the most frequent words of a given corpus, hence, the not so frequent words of the corpus will be used to identify the topic of a text. But, in style based text classifications, most frequent words of an author's writing are used to represent his/her style of writing. For this reason, rather than the most frequent words, masking the infrequent words of a writing to remove topic information, which are not related to the personal style of an author will be effective.

2.8.2 Feature Extraction and Selection in Authorship Attribution

In authorship attribution, the text is not the only imperative thing but the features that define characteristics of a writer and stylometry are more important. The study of stylometric features (stylistics) shows that people can be recognized by their relatively steady writing styles. In a given writing, the terms used, the selection of special characters, and the composition of sentences are used to define the style of an individual [62]. Earlier studies on authorship attribution, under diverse labels and criteria, have proposed catalogs of features to quantify the writing style, known as style markers [3].

However, studies in literature show that there are no such features set optimized and applicable to all people and to all domains. Chinchu J. et al., [64] claimed that features and their extraction process are very dependent on the text language. These features can be used to understand the uniqueness of an author's writing. These features are extracted from the author's text.

2.8.3 Stylometric Features

The key to inclosing the task of authorship attribution as a text classification problem is the selection of the feature sets that best describe the style of the authors. In the literature, a number of stylometric features are defined: lexical, character, syntactic, semantic and content-specific features.

Lexical Features

A straightforward and characteristic way to view a text is as an arrangement of tokens assembled into sentences, with each token corresponding to a word, number, or punctuation mark. An individual's preference on how to use characters and words in a sentence are learned using lexical features [62]. These lexical features may include frequency of individual alphabets and special characters, capital letters used in the beginning of sentences, total number of upper case letters, average number of characters per word or average number of characters per sentence. Creating lexical features that are effective discriminators of texts under study is one of the basic issues in an authorship analysis research.

In majority of authorship attribution studies, lexical features are used, at least moderately, to represent style [3]. In style-based text classification, the most common words (prepositions, pronouns, etc.) are found to be one of the top features to distinguish between authors [69] commonly known as "function words". In topic-based text classification methods, function words are usually excluded from the feature set. Essentially, function words are topic-independent and are used unconsciously by the authors. In any case, the determination of the specific function words that will be used as features is ordinarily based on subjective criteria and requires language-dependent skill [3]. There are different strategies to characterize a lexical feature set for authorship attribution, each with its own advantages and shortcomings. The basic one is to extract the most frequent words from the available corpus (i.e., word based-features), which needs the availability of complex tools like tokenizer, lemmatizers and stemmers [3]. The bag of words approach, which disregards word order information and the word n-grams approach [70], which captures content specific information rather than stylistic information, are the proposed methods to extract textual features in the literature.

Character Features

In character based feature extraction, a text is viewed as a mere sequence of characters defining various character level measures. These include alphabetic characters count, digit characters count, letter frequencies, punctuation marks count, uppercase and lowercase characters count, and so on [3]. Extracting frequencies of n-grams on character level capable of capturing nuances of style,

including lexical information, hints of contextual information, use of punctuation, capitalization etc., is the simplest approach in character features. This type of character features are also tolerant to noise [3]. The procedure of extracting the most frequent n-grams is language-independent and requires no special tools. However, [71] claims, in comparison to the word-based approach the dimensionality of this representation is considerably increased. This is because to represent a single long word several character n-grams are needed and character n-grams tend to capture redundant information. So far, character n-grams have been applied for style based studies by many researchers. Nevertheless, defining the appropriate value for n, how long the strings should be, remains an issue. As per [3], a large n is better in capturing lexical and contextual information, but it also captures thematic information. Moreover, a large value for n means a considerably bigger dimensionality of the representation. On the other hand, a small value for n capable to represent subword (syllable-like) information, but it is not sufficient enough to represent the contextual information. This makes the selection of the best n value a language-dependent procedure.

Syntactic Features

So far, syntactic language models have been used as an alternative to n-gram language models for machine translation, syntactic analysis and tree linearization [72]. Syntactic models are mostly preferred for their capability in capturing structural information, and can effectively improve long-range dependencies, fluency of large constituents and overall sentential grammaticality. The notion behind syntactic features is that authors have a habit of unconsciously using similar syntactic patterns. Therefore, compared to lexical information, syntactic information is considered more reliable authorial fingerprint [3].

Syntactic information requires robust and accurate NLP tools [3] which are capable of performing syntactic analysis of texts. The need for this NLP tools makes the extraction of syntactic measure a language dependent procedure, i.e., since it relies on the availability of a parser able to analyze a particular natural language with relatively high accuracy. Moreover, parsers tend to make unavoidable errors making such features produce noisy datasets.

Semantic Features

Semantic features, just like syntactic features, are higher level features which require deeper linguistic analysis. In feature extraction [3], the more detailed the text analysis for extracting stylometric features is the less accurate and noisier the produced measures will be. It is for this reason that many works avoided applying semantic features to address the problem of authorship attribution.

Sentence length features

It is obvious that some writers love to write in short sentences, while others prefer to write long blocks of text consisting of many clauses [73]. So far, using sentence length to represent an author's style of writing has not been given much of an attention for sentence lengths are considered much closer to the fully conscious end in the scarcely conscious to fully conscious spectrum of author's consciousness [74] . In books on the Amharic language grammar and writing like "አማርኛ ሰዋስው እና ሥነጽሑፍ" [28], it has been indicated that it is possible to distinguish authors of Amharic writings by the length of the sentences that they use in their writing. According to [28], in the history of Amharic literature, the author of the book fiker eske meqaber (ፍቅር እስከ መቃብር) by Hadis Alemayehu has been known by his preference to use longer sentences in his writings. This agrees with Stamatatos's [3] claim that an author would modify the length of his sentences according to tone, subject matter, the character of his various narrators and rhetorical purpose. In the literature, studies like [74] implemented the notion of sentence length patterns for the authorship attribution task through different techniques and algorithms.

Application Specific Features

All the above mentioned features (lexical, character, syntactic, or semantic features) are application-independent since they can be extracted from any textual data given the accessibility of a suitable processing tool and the availability of resources required for their measurement [3]. In an authorship analysis study one can define application-specific features to better represent the nuances of style in a given text domain. In more detail, for texts with certain topics and same genre, it is possible to define certain words that are frequently used within that topic or genre. Content-specific keywords can be used to capture properties of an author's style within a particular text

domain. Application specific features can also be defined depending on the type of language which will only be applicable to that specific language. For authorship attribution studies in languages with richer morphology where many function words are represented by prefixes and suffixes, morphological analysis has shown to be useful [12].

Feature Extraction and Selection

An important stage in authorship attribution is a process of finding distinctive features from the dataset which exhibit the writing style of each author individually. In authorship attribution, extracting features is a critical stage since it aims to determine unique features. After feature extraction, a feature selection process is applied to limit the dimension of irrelevant selections. Feature selection involves getting rid of attributes which fail to meet a given condition, consequently creating a subset or combination of the total number of features. More often, the feature sets used in authorship attribution studies are a combination of several types of features [3]. However, features like lexical and character features can significantly increase the dimensionality of the feature set. In such cases, feature selection algorithms like dimension reduction are applied to reduce the dimensionality of the representation.

So far, many feature extraction and selection techniques are proposed by different researchers in different studies. Forsyth [75] compared feature (character n-gram) sets selected by frequency with feature sets selected by distinctiveness and concluded features selected by distinctiveness more accurate. Information gain [76] is another feature selection algorithm which individually examines the discriminatory power of features. Chi-Squared [76], is a statistical feature selection method which measures divergence from the expected distribution assuming that feature occurrence is independent of class value, Tf − idf (term frequency – inverse document frequency) [17], frequency-based feature selection with odds ratio [77], and feature selection algorithm using discrimination information [78] are some of the feature selection techniques which are discussed in the literature. Feature extraction can also be used to reduce dimensionality of feature sets. The most traditional feature-extraction technique in authorship attribution studies is the principal components analysis, which provides linear combinations of the initial features [3]. Another is sparse principal component analysis (SparsePCA) [79] which is a variant of principal analysis which allows working with sparse data.

2.8.4 Authorship Attribution Approaches

In previous studies, the authorship attribution problem of several classes have been addressed in a variety of approaches. All these approaches are categorized in a different way by different researchers. [12] divides these approaches into 3 classes: (a) *unitary invariant approach*, in which a single numeric function of a text is sought to discriminate between authors; (b) *the multivariate analysis approach*, in which statistical multivariate discriminant analysis is applied to word frequencies and related numerical features; and (c) *the machine learning approach*, in which modern machine learning methods are applied to sets of training documents to construct classifiers that can be applied to new anonymous documents. Stamatatos [3] discriminates the authorship attribution approaches according to whether they treat each training text individually (profile based) or cumulatively (instance based) per author.

A. Profile Based Approaches

In this approach, all the available training texts of an author are concatenated into one single text file. The properties of the author's style are extracted from this big single file. A new text file is then compared with each author file, and the most likely author is estimated based on a distance measure [3]. Some AA studies concatenate all training texts of an author creating one big text file and extract a collective representation of that author's style (the author's profile) from this text file. The profile based approach ignores the differences that exist between texts written by the same author. Figure 2.1: shows the architecture of profile-based approach.

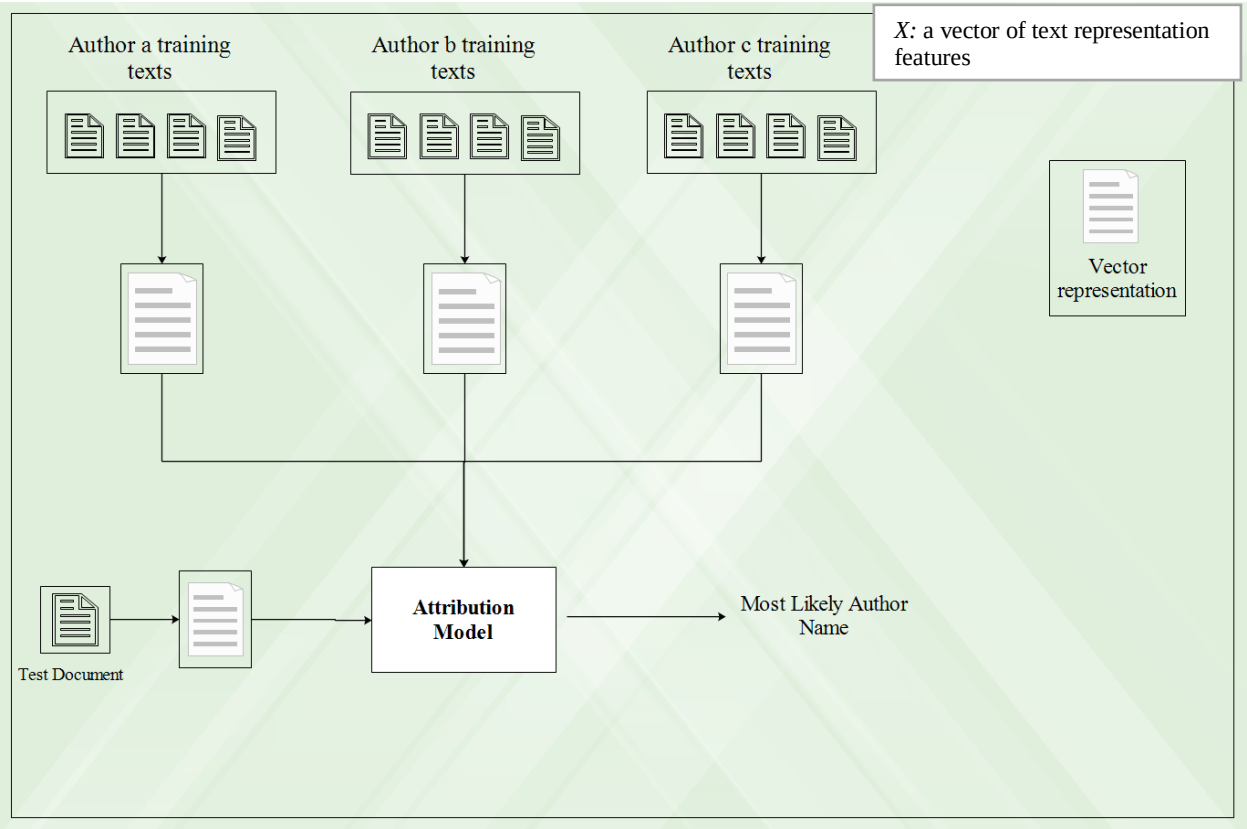

Figure 2.1: Architecture of profile-based approaches

In the training phase of profile based approaches, extraction of profiles for each candidate author and the attribution model is commonly done using a distance function that computes differences between the profile of each author and an unseen text. Stamatatos [3] describes how the profile based approach can be realized by using probabilistic and compression models.

Probabilistic models: this method works by maximizing the probability P(x|a) for a given text x to belong to a candidate author a. Then, the attribution model seeks the author that maximizes the following similarity metric:

$$author(x) = \arg\max_{a \in A} \log_2 \frac{P(x|a)}{P(x|\bar{a})}$$

Where the concatenation x_a of all training texts of the author a and concatenation of the rest of all the texts are used to estimate the conditional probabilities.

Compression models: [3] describes compression-based approaches which follow the profile-based methodology as the most successful of the compression methods. At first, x_a, a large file, is

formed from a concatenation of all the available texts for the i^{th} author which will then be compressed to produce a smaller version of it, $C(x_a)$, using a compression algorithm. At this point, to each text x_a the unseen text x is added. The compression algorithm will then be called again for each $C(x_a + x)$. The likeness between the unseen text and each candidate author is shown by the difference in bit-wise size of the compressed files $d(x, x_a)=C(x_a + x)-C(x_a)$.

Common n-grams: the common n-gram method uses a concrete representation of the author's profile. In particular, the profile of a text x (PR(x)) was composed by the L most frequent character n-grams of that text. The following distance is, then, used to estimate the similarity between two texts, x and y:

$$d(PR(x), PR(y)) = \sum_{g \in P(x) \cup P(y)} \left(\frac{2(f_x(g) - f_y(g))}{f_x(g) + f_y(g)} \right)^2$$

Where g is a character n-gram, $f_x(g)$ is the relative frequency of occurrence of the n-gram in text x and $f_y(g)$ is the relative frequency of occurrence of the n-gram in text y. Here, the dissimilarity between the two profiles is computed through calculation of the relative difference between their common n-grams. All the n-grams of the two profiles that are not common contribute a constant value to the distance. In common n-gram methods there are two important parameters that should be tuned: the profile size L and length of the character n-gram n [3].

B. Instance based Approach

In modern authorship identification approaches, each training text sample is considered as a unit that separately contributes to the attribution model [3]. In instance based approaches, each text sample of the author from the training set is represented by a vector of attributes (x). A classification algorithm is trained using these instances of the training set to develop an attribution model. Then, this developed model will be used to identify the true author of a given an anonymous text. Figure 2.2: shows architecture of instance-based approach.

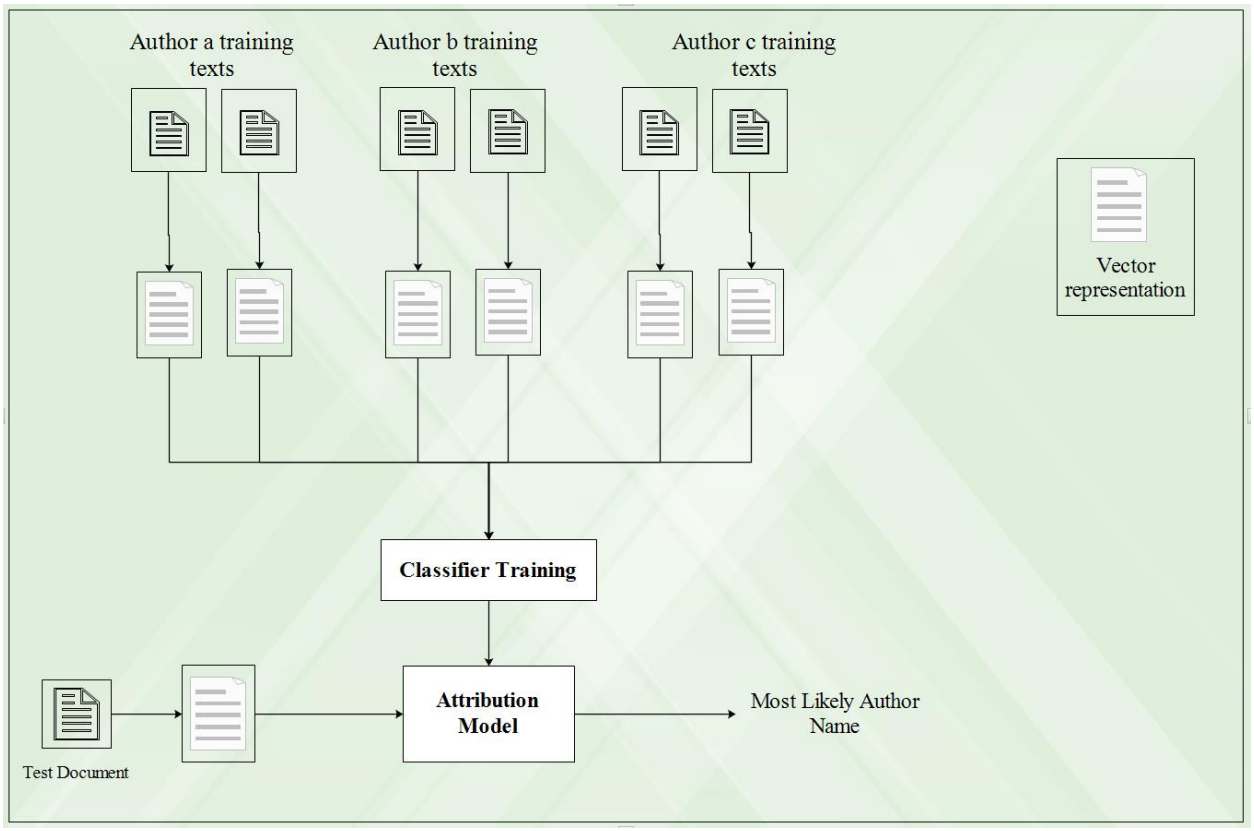

Figure 2.2: Architecture of instance-based approaches

In the literature, there exist various instance based approaches: vector space models, similarity based and Meta learning models and among them is the vector space models that comprise the majority of instance based approaches.

Vector space models

By representing the training texts in a multivariate form, it is possible to consider each text as a vector in a multivariate space. Then, a classification model can be built using a selection of statistical and machine learning algorithms such as:

- Discriminant analysis [80],
- decision trees [81],
- support vector machines [82],
- Classifier ensemble methods [83], and so on.

Similarity based models

A similarity based model computes pairwise similarity measures between the new text file and all the available training texts. Then, a nearest-neighbor algorithm is used to assess the most likely author for the unseen text. Various approaches of the similarity based model are proposed from the research community like the Delta on [84] and compression-based approach [85] .

Meta learning models

In addition to the general-purpose classification algorithms, one can design more complex algorithms specifically designed for authorship attribution in which existing classification algorithms may serve as a tool in this scheme [3]. The unmasking method proposed by Koppel et al. [86], which involves no training phase, is an interesting example of the Meta learning model.

Hybrid Approaches

The hybrid approach derives its basics from both profile based and instance-based approaches [87]. In this approach, as it is with the instance based approaches, all the training text samples are individually represented. However, as it is with the profile-based approaches, the vector representation for the texts of each author are averaged feature-wisely and as it is in profile based approaches, a single profile vector for each author is created. Then, the distance between the profile of an unseen text and the profile of each author is calculated to identify its rightful author.

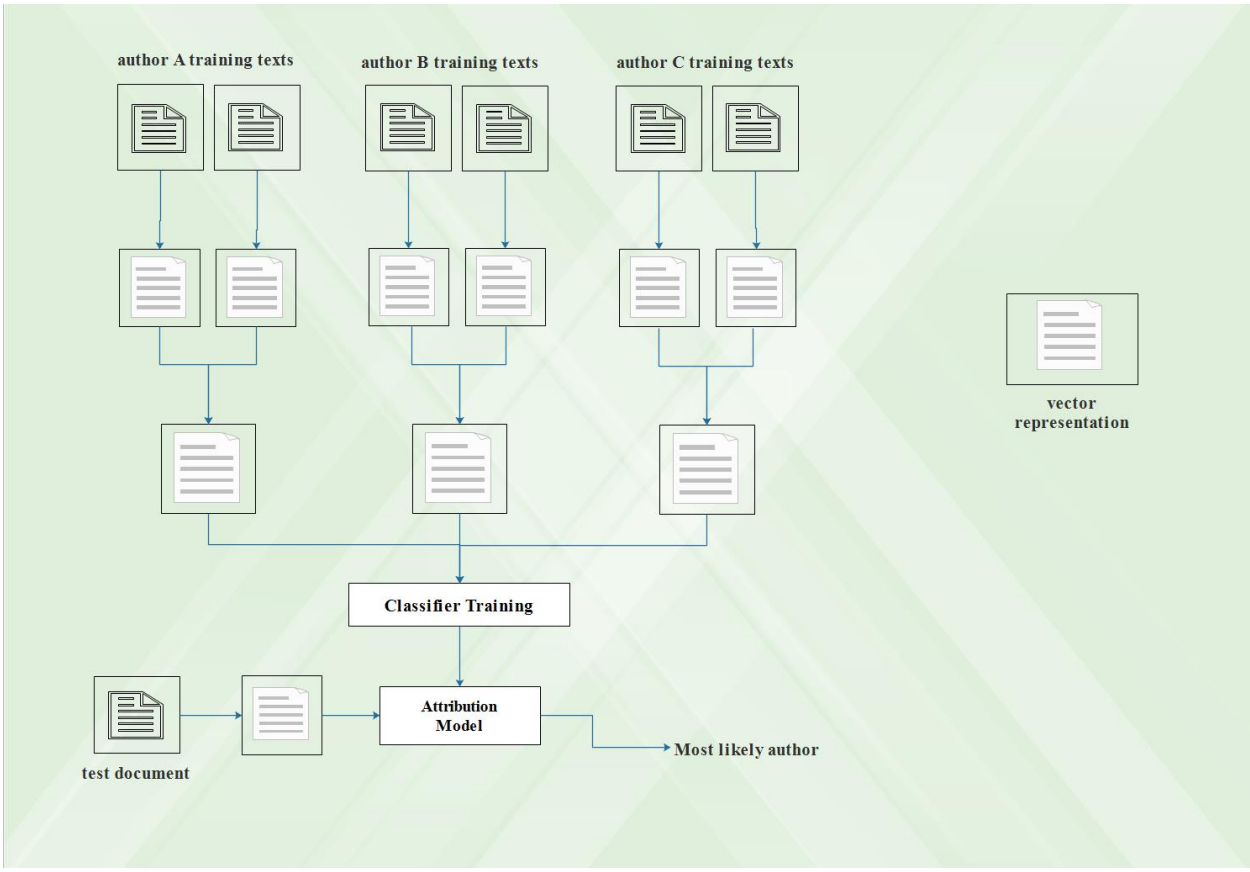

Figure 2.3: Architecture of hybrid approaches

C. Other Approaches to Authorship Attribution

Other than the above mentioned approaches to the authorship attribution problem, there are some approaches that are proposed by the research community. Among these are, *Graph based approach*, applied for author verification by Helena G. et al. [88], which defines a model to represent texts by means of a graph and extracts features from these graph to compute similarity. The *document weight approach* [66] in which case document weight is used to represent the document vector instead of using features or terms in the document. The other approach is using the concept of networks to the authorship attribution problem. According to [89], the concepts and methods in complex networks can be used to analyze texts at their different complexity levels. In this context, texts are represented as networks, where both nodes and edges may represent distinct textual aspects. Vanessa Q. et al. [90], Diego R. [91], and Shibamouli Lahiri and Rada Mihalcea [92] applied the concepts of networks to study stylistic properties of documents.

2.8.5 Classification Algorithms (Classifiers) in Authorship Attribution

There have been used many types of classification algorithms in several authorship attribution studies. So far, the most common type of classification algorithms used are the Naïve Bayes, decision trees, neural networks and support vector machine. Each of these classification algorithms perform far more differently from one AA study to the other given the type of language, features and method selected in that particular study.

Support Vector Machines

Support vector machines (SVM) are the most successful type of classifiers in the pattern recognition and classification, regression and outlier detection studies [82]. A support vector machine algorithm works by constructing a hyperplane or set of hyperplanes in a high-dimensional space to separate the classes. The hyperplane is used to maximize the margin between the points in the space. This allows the model built to predict the target class for new examples. The SVM can be used to solve both linear and non-linear problems. The SVM is mostly preferred for its effectiveness in high dimensional space, versatility and its capability to deal with cases where the number of dimensions is greater than number of samples [93].

Naïve Bayes

The Naïve Bayes (NB) method is based on the concept of the 'Bayes' theorem which is the naïve assumption of conditional probabilities between every pair of features [94]. Naïve Bayes is one of the simplest of classification algorithms. There are multiple variations of the Naive Bayes algorithm: the gaussian NB, multinomial NB, bernouli NB, complement NB and out-of-Core NB model fitting. Each of these algorithms differ mainly by the assumptions they make regarding the distribution function [93]. Even though they are simple and very fast in making predictions, Naive Bayes algorithms face the "Zero Conditional Probability Problem" and the naïve assumption is considered a disadvantage [94].

Decision Trees

Decision trees are tree like structures which usually are similar to flowcharts where internal nodes are used to represent features, branches are used to represent decisions and an outcome is represented by a leaf node. Decision trees are commonly used in machine learning, statistics and data mining for predictive modeling. There are different decision tree algorithms [95]. Classification and regression tree, Chi-square automatic interaction detection and Conditional

inference trees are amongst them. Decision trees are simple, can capture non-linear patterns and are effective with large datasets but are vulnerable to overfitting, perform poorly in high dimensional feature spaces and cannot always guarantee optimal decision tree. In authorship attribution studies, decision trees are not commonly used for classification like the SVM but are applied on studies like [96].

Neural Networks

Neural networks are defined as circuits or networks of neurons. Neural networks are artificial neurons capable of solving artificial problems inspired by the data processing of biological neural systems. Neural networks as a classification method, apart from their computational requirement and overfitting problem, are: data driven self-adaptive methods, can approximate any function, are capable of estimating posterior probabilities and are non-linear models [97]. There are various types of neural networks. Recurrent neural network, modular neural network, convolutional neural network and radial basis function neural network are some of the neural network types. They have been applied in various applications such as regression analysis, classification tasks and data processing like clustering. Neural networks have been used in many text classification tasks and have also been applied in authorship attribution studies [98].

2.8.6 Evaluation of authorship attribution systems

Evaluating an authorship attribution system involves measuring the overall discriminatory power of the system. According to [3], to test the performance of an attribution system and for the evaluation method to be most appropriate, it is crucial for the attribution methods to be robust with a limited amount of short texts. Besides, the system should be tested using a balanced test corpus. In the literature, the most common type of evaluation metrics used are accuracy, precision, recall and F-score [99].

- **Accuracy**: is the percentage of the classifier's correct predictions. It is the ratio of correct predictions to total predictions.

$$Accuracy = TP+TN/TP+FP+FN+TN$$

- **Precision**: is the ratio of classifier's correct positive predictions to the total positive predictions.

$$Precision = TP/TP+FP$$

- **Recall**: is the ratio of correctly classified positive predictions to the number of positive predictions in the data.

  $$Recall = TP/TP+FN$$

- **F-score**: is the harmonic mean of precision and recall.

  $$F\text{-}Score = 2*(Recall * Precision) / (Recall + Precision)$$

Where, TP = True Positive, TN = True Negative, FP = False Positive and FN = False Negative. The results of these evaluation metrics methods show the assessment of how well a classifier works.

Chapter Three: Related Work

3.1 Introduction

Authorship attribution is one of the most extensively studied authorship analysis problems in the field of text classification. Many researchers in the research community studied the problem to develop systems to correctly attribute anonymous texts using different features and classification methods for several languages. In this chapter, we present a review of some related works on authorship attribution. These works are categorized into sections based on the language they are developed for, i.e., English, Arabic and other languages (Malayalam, Lithuanian, Portuguese and Punjabi Language) which are all studied using different approaches.

3.2 Authorship Attribution for English

Sindhu R. et al. [100] presents an approach for authorship attribution using probabilistic context-free grammars which involves building a probabilistic context-free grammar (PCFG) for each author and using this grammar as a language model for classification. The approach involves, given a certain set of training documents from different authors, a PCFG is built for each author based on their documents. Then, a test document is parsed using each author's grammar and assigned to the author whose PCFG produced the highest likelihood for the document. A statistical parser trained on a generic corpus is used to automatically annotate the training documents for each author. The dataset is created from a variety of documents with known authors including news articles on a wide range of topics and literary works like poetry. A total of 5 different datasets: football, business, travel, cricket, and poetry are considered. For testing the approach, 15 documents per author for datasets with news articles and 5 or 10 documents per author for the poetry dataset were used. The authors compared the performance of the proposed approach to bag-of-words classification and n-gram language models. For the task of authorship prediction, they hypothesized that the frequency of specific stop words could provide useful information about the author's writing style and went on to verify the hypotheses with initial experiments that removing stop words degraded performance. In order to utilize both syntactic and lexical information, PCFG-E (an ensemble using a PCFG model, the bag-of-words MaxEnt classifier, and an n-gram language model) is used. Thus, the authors assert using syntactic information alone is generally not much accurate than using n-gram concluding that both syntactic and lexical information are useful in effectively capturing authors' overall writing style.

Mechti et al. [101] proposed a hybrid method that combines a set of stylistic and statistical features in a machine learning process. The proposed method is based on the combination of statistical and stylistic features using the SVM algorithm. The authors implemented the method using a system they developed known as HyTAI (Hybrid Tool for Author Identification). NLP tools from the Apache OpenNLP library are used to extract stylistic and statistical features. Frequency of lexical features and average sentence length features are also considered in this study. To test the method, the authors used the corpus of the PAN'@ CLEF'2015 conference (a corpus consisting of 200 collections of English documents which include 518 known texts and 200 unknown texts). The method is evaluated using the c @ 1 measure for multiple classifiers (SVM, Bayesian Networks, Naive Bayes, Decision tables, Decision tree and KNN). The best results the authors obtained for the proposed hybrid system is using the SVM algorithm which is equal to 0.59 accuracy rate for c@1.

3.3 Authorship Attribution for Arabic

Authorship attribution in Arabic poetry using machine learning is an attribution model presented by Ahmed et al. [102]. The model is designed to apply the Naïve Bayes (NB) and support vector machine (SVM) classification algorithms together in old Arabic poetry. In this work, the authors identified Rhyme and Meter as the characteristics of old Arabic poems that distinguish them from the rest of literary. The features: Character, Poetry Sentence Length, Word Length, Rhyme, Meter and first word in the sentence are extracted and used as a basis in creating the training and test sets. Features are selected using the feature selection methods: chi-squared (x^2) and information gain (IG). The datasets used are grouped into training dataset with known poets and test dataset with unknown poets. The model is experimented using both the Naïve Bayes and SVM classifiers on 73 randomly selected samples of 73 authors. The model has been evaluated for each feature extracted and recorded the best accuracy of 98.63% using the SVM classifier for character and word length features.

Abuhaiba and Eltibi [6] addressed author attribution of Arabic texts using extended probabilistic context free grammar (XPCFG) language model through lexical and syntactic features. The method uses language models to assign an author to a test document of an unknown author. This method begins with creating a language model, a model that captures the language syntax of the author which is done for each author from his own training documents. The syntax features

extracted are syntactic features which are capable of capturing an author's writing style. Other than the syntax, the model involves more syntactic features like the part of speech tagging (POST) and lexical features. The algorithm uses a parsing tool to generate an XPCFG model for each author from a set of training documents. The rules generated here involve two different sets, terminal rules, and non-terminal rules. For each production rule generated, a probability is computed. Then a score is computed for each rule to compute the dependency between this rule and its corresponding author, which is accomplished by computing the chi-square (x^2) score for each rule. Since the model contains different features, optimum weights are found using a genetic algorithm to govern how each feature participates in the classification. Error rate is used to measure the efficiency of the classifier. The dataset used contains over nine authors, 20 Arabic documents each. The training and testing is done using the leave-one-out method. The initial error rate of the system is 20.6%. The authors claimed that the system depends on the rules generated by the parser since the parser has some limitations as it cannot split clitics resulting in inaccuracy for some sentences.

3.4 Authorship Attribution for Other Languages

Applicability of relevant authorship attribution technique in Malayalam by Shabeeb PK [103] is a work which presents an authorship attribution system for Malayalam transcripts based on the n-gram model. The study mainly focuses on adopting widely used authorship attribution techniques implemented in English to Malayalam, which the author claims is comparatively a complex and agglutinative language making it very different from other inflectional languages such as English/European languages. In this work, given the lack of effort made to implement authorship attribution into languages other than English and a few others, the language barrier has been identified as a major challenge of authorship attribution. This model uses n-grams for representing the text. To exploit the complementary nature of character and word level information, a vector of both character and word n-grams of different sizes are combined. The dataset is constructed from a collection of corpus written by possible authors which is divided into training and testing set. The model is tested with a defined threshold of 75% resemblance as the threshold for classifying texts. From the experimental results, the paper concludes that optimal results are achieved when both training and testing sets for authorship attribution contain merged transcripts indicating the text-size effect in the authorship attribution accuracy correlates directly with the test set.

Silva et al. [104] compared the robustness of several types of stylistic markers to help discriminate authorship at sentence level. The study investigates authorship analysis in Portuguese texts at sentence level stating that performing authorship analysis at this level raises the additional problem that style markers should be able to work with short text strings and intra-sentence information only. POS-based features, pronouns, punctuation, length, suffixation, and conjunctions, which are considered by the authors as content agnostic features and potential markers of authorship, are extracted at sentence level. The authors built a corpus of editorials and opinion articles posted by columnists of a Portuguese daily newspaper. The corpus built comprises 915 posts by 23 commentators. SVM is used as the classification algorithm for its robustness in text classification tasks. The experimental results presented show that the performance they obtained using any of the subgroup of markers alone is lower than the performance obtained using all the stylistic features. The study concludes by claiming content-agnostic features can effectively be used for authorship analysis at sentence level. Punctuation is also identified as the most robust stylistic features analyzed, affixes and pronouns are identified as not enough to perform sentence-level authorship attribution with robustness, and word and sentence length obtained similar results to punctuations.

Utka et al. [7] presented an authorship attribution study which reports authorship attribution results based on the exploration of the effect of the author set size when dealing with normative and non-normative Lithuanian language texts and using supervised machine learning techniques. This work mainly focuses on determining how authorship attribution results declines while the number of candidate authors increases: i.e. starting from 3, going up to 5, 10, 20, 50, and 100. The study is also aimed at investigating the influence of different features (lexical, character, morphological, etc.) and language types (normative parliamentary speeches and non-normative forum posts through supervised machine learning techniques). Two supervised machine learning approaches: Support Vector Machine (SVM) and Naive Bayes Multinomial (NBM) were explored. Two datasets, **ParlTranscr and LRytas,** were used to generalize findings over different domains and language types. **ParlTranscr**, contains unedited transcripts of parliamentary speeches and debates, representing formal spoken but normative Lithuanian language. **LRytas**, contains forum data made up of 11 general topics which is full of informal words, emoticons, foreign language insertions, word shortenings and diacritic eliminations, representing the informal non-normative Lithuanian language. Experiments were carried out with the stratified 10-fold cross-validation and evaluated

using accuracy and micro/macro average F-score metrics. The authors used Chi-squared feature extraction method, SMO polynomial kernel with SVM and NBM implementations in the WEKA machine learning toolkit. Because they exceed random and majority baselines, results were claimed reasonable and appropriate by the authors for solving the task and they preferred SVM as a much better selection since it outperformed NBM for most of the cases, except for a couple of cases when both methods achieved the same accuracy. It is presented that with the same number of candidate authors, the accuracy of **LRytas** was much lower than **ParlTranscr**. As the authors state, this was due to the language type, text length, and training dataset size. The authors also stated that, in general, the best feature type for **ParlTranscr** (Normative) dataset is based on the lemma and part-of-speech information and the best feature types for **LRytas** (Non-Normative) dataset is based on the character n-grams. Since the Lithuanian language is highly inflective, morphologically and vocabulary rich; and to deal with the normative language; morphological tools were found to be helpful. It is presented that, using parliamentary data (thus normative Lithuanian language) the results exceed baseline by 62.7% and reach even 70.6% of accuracy with 100 of candidate authors and using forum posts (thus non-normative texts) results exceed baseline by 20.7% and reach 30.9% of accuracy.

Navinder et al. [105] applied the authorship attribution process on Punjabi poetry corpus consisting of Punjabi poetry written by 10 different poets. This system uses lexical features to identify the author of a given poem. Word n-gram and character n-gram feature are extracted using the tf-idf (term frequency- inverse document frequency) vectorization method and are used to train SVM classifier. The whole process consists of training a network with 856 poems whose authors are known. Then, an anonymous poem is given as input to the network which gives the most appropriate name of the author as output. Punjabi poetry corpus, a store having collection of poems related to a particular poet was used as a dataset. The authors used poetry of 10 different poets from websites to construct a poetry corpus of 865 poems for training dataset and 456 poems for testing dataset. The performance of the proposed system is evaluated with precision, recall, F-score and accuracy parameters. The authors tested char unigram, char bigram, char trigram, word unigram, word bigram and word trigram features. The paper presents a precision value of 81% was obtained from word unigram feature which was the best among all the features that have been employed in the experiment. An accuracy of 79% was obtained by char tri- gram feature. The authors discussed that they have observed the character based features are better than the word

based features with 68% precision, 62% recall, 51% F-score and 65% accuracy. However, the poetry corpus used in the study is limited only to 10 poets.

3.5 Summary

In this chapter some studies related to the problem of authorship attribution are reviewed and discussed. The studies are reviewed based on the language they are developed for. Even though some works are studied in few languages, several works on the problem of authorship attribution focus on the English language. These studies are implemented using different statistical and machine learning techniques. Numerous type of stylistic and statistical features, feature extraction methods and classification methods (classifiers), are proposed and used. In the task of authorship attribution, the feature types used, the method employed, the classifier used and the language the study is carried on has a direct effect on the effectiveness and performance of the system developed. For the same language different features can result in different performance. A study on the same language with similar features but using different corpus leads to different findings. Other than that, the nature of languages by itself, like the complexity of one language to another, makes the study of authorship attribution far more challenging. Even though some summarization, topic modelling, and named entity recognition systems have been developed for Amharic language, the problem of authorship attribution is not yet studied for the language. In this study, we addressed the problem of authorship attribution through machine learning techniques using the combination of both profile and instance based approaches.

Chapter Four: Design of a Model for Authorship Attribution of Amharic Documents

4.1 Introduction

The authorship attribution problem is considered a demanding text classification assignment since for a system to correctly predict the true author of a given anonymous text one must first prepare the desired corpus, extract and select the right set of features and choose the right classification algorithm. In this chapter, the proposed model for the attribution system is discussed in detail.

4.2 System Architecture

The proposed authorship attribution (AA) model for Amharic documents involves a training phase and an author attribution phase. In this study, the training data (Author Dataset) is used to create the author profile. First, the training data is preprocessed under preprocessing and features are extracted in the feature extraction process. A classifier training component is used to create an author learning model using a feature vector set for all authors in the training dataset. In the attribution phase, an anonymous text is passed through all the processes as it is in the training phase except there are no profiles that the classifier will be trained with. Instead, the features extracted will be directly fed into the classification component which compares it to the features of the candidate authors to make a decision. All the components of the system with their detailed function are explained in the next subsections of this chapter. The proposed system architecture is shown in Figure 4.1.

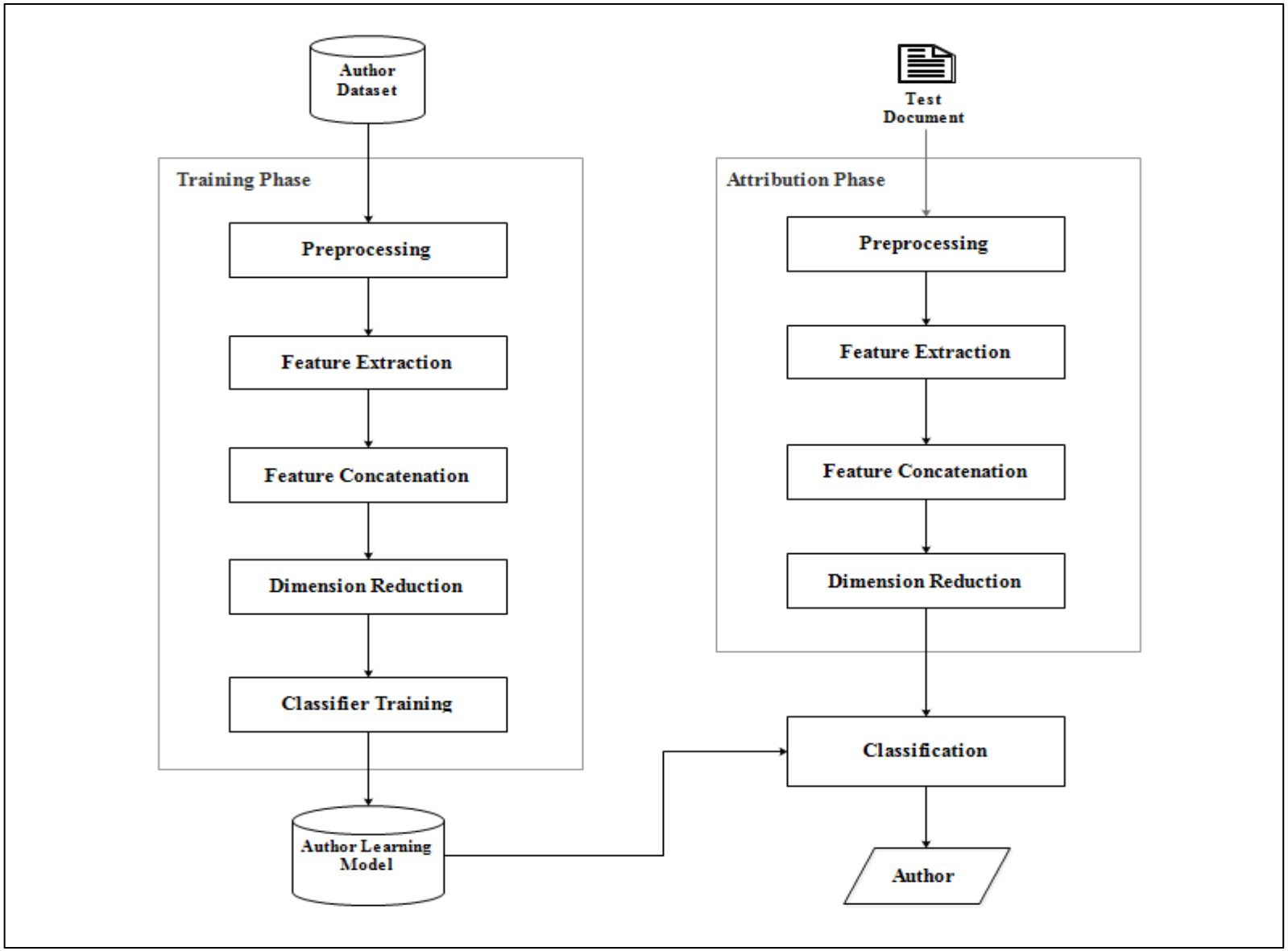

Figure 4.1: Model Architecture

4.2.1 Preprocessing

The task of text preprocessing is unavoidably useful process since it prepares all the documents of the training data and the test document for further analysis. The training data (Author Dataset) is a collection of poems and articles of different newspapers and magazines of authors. In the preprocessing phase, all the unnecessary elements are removed through the normalization subcomponent. The normalized texts will then be identified as a poem or a non-poem using the text identification component. Texts identified as a poem are versified through the poem versification component for further poem specific feature extraction where as non-poem texts are tokenized and are sent to the text distortion, part of speech tagging, Stemmer and sentence segmentation components so other features which are considered to be representatives of an author's style are extracted in the feature extraction stage. The normalized texts will also be directly fed to the space n-gram feature extraction component. The preprocessing task is identical for the training and attribution phases which means that both the training dataset and the test document have to pass through the same preprocessing procedures. Figure 4.2 shows the sub-components of the preprocessing component.

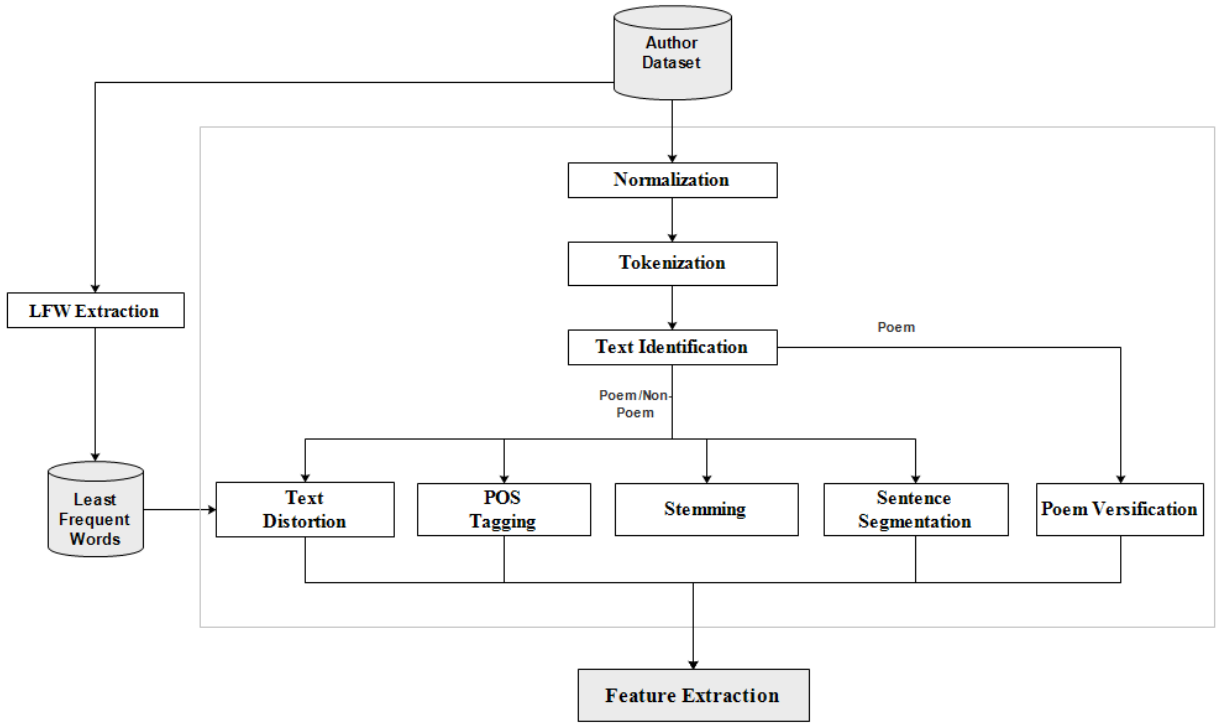

Figure 4.2: Preprocessing

Normalization

If the input data contains any irrelevant data which is not really necessary in extracting the features representing an author's writing style, the normalization component will be a key tool in changing the input text to a suitable one. In newspapers and magazines, articles usually contain names of authors for a particular published article, dates the article is published on, photographs related to the article, page numbers, page indicators if part of a single article continues on some other page, part numbers of the articles if the article is continued from earlier magazine or newspaper publications and other effects like animations which are not basically important in the attribution process. Multi-authored articles, an article written by multiple authors, and interview articles of authors which we considered as not useful in the training process are not included.

The output of this normalization component, which is a normalized text, will be a direct input to the tokenization component.

Tokenization

In the proposed system, tokenization will be used to create tokens of words and punctuation marks. As it is done in many other classification works, punctuation marks will not be used as boundary markers. Instead, tokens of punctuation marks will be created so they will later be used in the feature extraction process. For example, the sentence "ወፎችን ከሰማይ ላይ በወስፈንጠር መምታት፣አሳ ማጥመድ፣የዱር ማርና ፍራፍሬ መሰብሰብ እንዲሁም ትኩስ ወተት ከላሞች ጡት በቀጥታ መጠጣትን ተምሬያለሁ።።" will be tokenized into ['ወፎችን', 'ከሰማይ', 'ላይ', 'በወስፈንጠር', 'መምታት', '፣', 'አሳ', 'ማጥመድ', '፣', 'የዱር', 'ማርና', 'ፍራፍሬ', 'መሰብሰብ', 'እንዲሁም', 'ትኩስ', 'ወተት', 'ከላሞች', 'ጡት', 'በቀጥታ', 'መጠጣትን', 'ተምሬያለሁ', '።።']. The output of this tokenization component, tokens of words and punctuations, will be an input to the text identification component in preprocessing, and the punctuation n-gram feature extraction component in feature extraction.

Text Identification

A given text will be identified as a poem or non-poem before further preprocessing tasks are applied to it. The text identification component is used to identify whether a given input text is a poem or non-poem. The input to this component is a normalized text from the normalization component. If a text is identified as a non-poem it will be an input to text distortion, POS tagging, stemming and sentence segmentation components. However, if a text is identified as a poem it will be an input to all components listed earlier for non-poem texts and also to the poem versification

component so poem specific features will be extracted in feature extraction. To identify a text as a poem or non-poem we adopted shape features used in [106]. The attributes (average mean and standard deviation of line length, paragraph, punctuations and numbers in the text) are used. Gaussian probability distribution of these attributes are used to identify the texts.

Text Distortion

In our work, for the text distortion task, we adopted the techniques employed by Stamatatos [67]. First, instead of the most frequent words, we identified the least frequent words ($\mathbf{W_k}$) of our training data for a certain value of k. Most of these least frequent words happened to be function words which are basically retained in the preprocessing steps of topic based Amharic text classification studies. Then, these set of least frequent words are used to mask the not so frequent words in the training set.

On the other hand, an author's preference to use the number 2,000,000 instead of 2 ሚሊዮን (million) or 2 ሚ. And 3000 instead of ሦስት ሺህ (three thousand) or 3 ሺ. Are considered as a style choices of the author. The information contained in digits represent telephone numbers, dates, or values etc. which is mainly associated with the genre rather than the style of an author. Since the use of the format, not the exact values or the combinations of digits is required, digits are also masked along with the other characters. Punctuation marks, considered most effective style markers, are left unmasked.

Basically, this text distortion component masks every word which is an element of the least frequent set with an asterisk (*) and digits with (#) creating a new distorted form of the original text. This distorted form of the original text is then passed to the n-gram feature extraction component in the feature extraction process. Table 4.1 shows an example of the distortion algorithm

Table 4.1: Distorting a text using text distortion algorithm

Original text	ባለፈው ሳምንት በአዲስ አበባ የሥራ ጉብኝት ሲያደርጉ የነበሩት የአሜሪካ የውጭ ጉዳይ ሚኒስትር ሬክስ ቲለርሰን፣ በፕሬዚዳንት ዶናልድ ትራምፕ ማክሰኞ መጋቢት 4 ቀን 2010 ዓ.ም. ከቀትር በኋላ ተሰናበቱ። ፕሬዚዳንቱ በትዊተር ገጻቸው ላይ የሲአይኤ ዋና ዳይሬክተር ሆነው በማገልገል ላይ የነበሩት ማይክ ፖምፒዮ እንደሚተኳቸው፣ በፖምፒዮ ቦታ ደግሞ ጊና ሃስፔልን መሾማቸውን አስታውቋል።
Distorted text (k = 100,000)	ባለፈው ሳምንት በአዲስ አበባ የሥራ ጉብኝት ሲያደርጉ የነበሩት የአሜሪካ የውጭ ጉዳይ ሚኒስትር ሬክስ ቲለርሰን በፕሬዚዳንት ዶናልድ ትራምፕ ማክሰኞ መጋቢት # ቀን #### * ም ከቀትር በኋላ ***** ፕሬዚዳንቱ በትዊተር ገጻቸው ላይ የሲአይኤ ዋና ዳይሬክተር ሆነው በማገልገል ላይ የነበሩት ማይክ ፖምፒዮ ******** ***** ቦታ ደግሞ ** ***** መሾማቸውን አስታውቋል

Distorted text (k = 120,000)	ባለፈው ሳምንት በአዲስ አበባ የሥራ **** ***** የነበሩት የአሜሪካ የውጭ ጉዳይ ሚኒስትር *** ***** ******* **** ትራምፕ **** መጋቢት # ቀን #### * * **** በኋላ ***** ፕሬዚዳንቱ ***** **** ላይ ***** ዋና ዳይሬክተር ሆነው ****** ላይ የነበሩት *** **** ******** ***** ቦታ ደግሞ ** ***** ****** አስታውቋል
Distorted text (k = 200,000)	**** **** **** *** *** **** ***** ***** ***** *** *** ***** *** ***** ******* **** **** **** **** # ** #### * * **** *** ***** ****** ***** **** ** ***** ** ****** *** ****** ** ***** *** **** ******** ***** ** *** ** ***** ****** ******

Part of Speech Tagging

In a sentence, part of speech (POS) of a given word describes its function within the sentence. Part of speech tagging involves identifying the function of a word in the context of the sentence it exists in. In the proposed system, the process of part of speech tagging is carried out using an Amharic POS tagger which produces part of speeches of Amharic words in a sentence. We used a part of speech tagging tool, which performs automatic tagging of words in a sentences, developed by Tsegaye Andargie for the thesis work [107]. The input to this component is a tokenized text from the tokenization component. The output of this component, words with their corresponding part of speeches, is used as an input to the part of speech n-gram feature extraction component in the feature extraction stage.

Stemming

In natural language, stemming involves the reduction of inflected or derived words into their root or stem form. Its main advantage is shortening the vocabulary space of a dataset which improves the size of the feature space. Given the higher degree of inflection in the Amharic language, the use of stemmer to reduce words into their root form will be significant. In this study, we created a stem/lemma of each word from all author dataset using, HornMorpho, a morphological analyzer by [108]. For example, for the poem verse" ስወለድ ጀምሮ ኣብሮኝ የሚኖረው" the equivalent stemmed form will be "ወለደ ጀመረ አበረ ኖረ". Thus, these stemmed form of author's dataset is used to create character and word n-gram features in the feature extraction process.

Sentence Segmentation

Unlike texts in some languages for which no boundaries are explicitly indicated in the written language, in Amharic, boundaries of words and sentences are explicitly marked. This means that knowledge of abbreviations, collocations and sentence starters is not required to train the sentence

tokenizer to correctly predict sentence boundaries. Instead, the explicit sentence boundary markers, i.e. punctuation marks, are needed. In the Amharic language, the punctuation mark which signifies the end of a given sentence is "**::**" (Arat Netib). Therefore, tokenizing sentences in the language involves finding the punctuation mark needed to segment the text. The sentence segmentation task is done by splitting each article into sentences using the punctuation mark "**::**" as a split point or boundary marker. This will create a list of sentences of each article of all the author's writing making it ready for the sentence length feature extraction process.

Poem Versification/Lineation

A verse is the most important element of a poem in poetry. Even though a verse can be used to represent a group of words in a poetic structure it formally denotes (equivalent to) a single line of a poem. In poetry verses usually require stanzas (groups of verses) and rhymes. Versification/Lineation involves the process of segmenting poems into individual lines or verses. The input to this component are poems from the text identification component. The poem Versification component does the task of versifying normalized poems into verses making the poems ready for the extraction of poem specific features.

4.2.2 Feature Extraction

Feature extraction is a key component in an authorship attribution task since extracting the right set of features to represent an author's style plays the most important role in the attribution process. Authorship attribution studies are differentiated from one another by the features extracted, attribution methods used and the dataset the study is carried on. Given its importance, we have performed the feature extraction task through different n-gram feature extraction components. Separately, each component in the feature extraction process is discussed in detail with algorithms next.

N-Gram Features

N-grams are contiguous sequences of terms (words) with length N. N-gram features have been used in various NLP studies for various tasks including text classification. The simplicity and scalable nature of n-grams makes them effective. In our study, we proposed to devise the process of extracting n-gram features in five different methods: word n-grams, character n-grams, punctuation n-grams, space n-grams, and part of speech n-grams as the success of an n-gram classifier is believed to be dependent on how many distinctive n-gram features are available for

each author. Each of these methods will be able to extract the different types of n-gram features and are discussed in detail below.

A. Word N-gram Features

Word n-grams are a combination of n words. For example for the sentence "የምሺዞ መንደር ትራንስኪ ውስጥ በባሽ ወንዝ ዳርቻ ትገኛለች።" word bigrams (2-grams) could be "የምሺዞ መንደር", "መንደር ትራንስኪ", "ትራንስኪ ውስጥ", "ውስጥ በባሽ", "በባሽ ወንዝ", "ወንዝ ዳርቻ", "ዳርቻ ትገኛለች". The word n-gram feature extraction component accepts distorted texts from the text distortion component and extract word n-gram features. Since the least frequent words are masked through text distortion, the word n-gram features are extracted from the most frequent set of words of each articles of an author's writing and stemmed forms of the author's texts which are later used to create the author learning model along with other character n-gram features.

B. Character N-Gram Features

Character n-grams are contiguous sequences of n items (characters) from a given sample of texts. In the proposed system, the character n-gram feature extraction will extract patterns of frequency and count of character level features like alphabetic characters, digit characters, letters, etc. It is applied in two different ways:

- ✓ First, similar to the word n-gram feature extraction, the process is applied on a distorted form of the original text for each article and stemmed forms of each author's texts.
- ✓ Second, the process is applied on the different sets of sentence categories created for each article based on sentence lengths.

For the latter, the categories are created based on sentence lengths with the assumption that some writers love to write in short sentences, while others prefer to write long blocks of text consisting of many clauses and also an author would modify the length of his sentences according to tone, subject matter, the character of his various narrators and rhetorical purpose. For these reasons, to capture sentence length patterns in each of the articles, we opted to extract character n-gram features from these set of sentence categories. For our study, we modified one of the techniques employed by [74], which groups sentences of a particular essay according to their length, to articles defining new sentence length groups for each article. Based on the sentence categories defined on [109], we defined five groups of different categories depending on their sentence length: Staccato,

Telegraphic, Short, Medium and Long. Each category contains groups of sentences with different lengths.

- **Category 1 – Staccato (c1)**: this category consists of sentences with not more than two words.
- **Category 2 – Telegraphic (c2)**: this category consists of sentences with not more than five words.
- **Category 3 – Short (c3)**: this category contains sentences with words between five and ten.
- **Category 4 – Medium (c4)**: this category contains sentences of words between fifteen and twenty.
- **Category 5 – Long (c5)**: this category contains all the sentences holding more than thirty words.

To leave no sentences uncategorized we redefined categories 4 and 5 to hold sentences of lengths between 10 and 20 words, and above twenty words respectively. The input to this component is segmented sentences of each article from the sentence segmentation component in the preprocessing phase. It will take list of sentences of each articles of an author, categorizes them into the five categories defined based on their length and stores the sentences in each category. Each category c_i will contain sentences in the range defined for the category c_i. Algorithm 4.1 shows the sentence length and sentence categorizing computation.

```
Algorithm: Sentence categorizing Pseudocode
Input: S : List of sentences in article A from sentences segmentation component
Output: Sentences of article A in five categories
Begin:
      c1, c2, c3, c4, c5 = [ ], [ ], [ ], [ ], [ ]
      For each sentence s in S:
          temp = length of s # length of sentences is number of words in it
          If temp <= 2:
              Append sentence s to c1
          else if temp > 2 and temp <= 5:
              Append sentence s to c2
```

```
            else if temp > 5 and temp <= 10:
               Append sentence s to c3
            else if temp >= 10 and temp <= 20:
               Append sentence s to c4
            else:
               Append sentence s to c5
            End if
      End for
Return c1, c2, c3, c4, c5
End
```

Algorithm 4.1: Sentence categorizing Pseudocode

Once all the sentences are categorized into the five categories, character n-gram features are extracted from these sets of categories.

C. Punctuation N-Gram Features

As different authors use punctuations differently, the punctuation n-gram feature extraction component will identify and capture patterns of punctuation mark usage of an author in many of his writings. The input to this component is a tokenized form of each article from the tokenization component. All the tokens in each article is scanned for punctuation marks and the tokens with a punctuation mark are retained while those tokens with no punctuation mark are removed. Then, the tokens retained are rejoined for n-gram feature extraction. The punctuation n-gram feature extraction extracts n-grams through four different categories: beginning punctuation n-gram, middle punctuation n-gram, end punctuation n-gram and whole punctuation n-gram. With the exception of the whole punctuation n-grams, all the three punctuation n-grams are extracted for n values greater than or equal to 3.

- *Beginning Punctuation n-gram*: n-grams for which the punctuation mark is the first character.
- *Middle Punctuation n-gram*: n-grams for which the punctuation mark is the middle character.
- *End Punctuation n-gram*: n-grams for which the punctuation mark is the last character.

- *Whole Punctuation n-gram:* n-grams for which the punctuation mark is any of the characters in the n-gram.

Algorithm 4.2 shows extraction of punctuation n-grams. Table 4.2 shows an example of punctuation n-grams for n=3.

```
Algorithm: Punctuation n-grams extraction Pseudocode
Input:  Tokenized text t of article A from tokenization component
        Amharic Punctuations P
Output: beg_punct, mid_punct, end_punct, whole_punct
Begin:
      punct_tokens, ngram_tokens = [], []
      beg_punct, mid_punct, end_punct, whole_punct = [], [], [], []
      For token t in article A
         For a character c in t
            If c in punctuations P:
              append the token to punc_tokens
            End if
         End for
      End for
     # create n-grams
     # Compute ngram-tokens from punct_tokens
     For n-gram n in ngram_tokens
         temp = n
         append temp to whole_punct
         For character c in temp
           If c in punctuations P:
             If the character c is the first element in the n-gram temp
                 append n-gram temp to beg_punct
                 break
             Else if the character c is the last element in the n-gram temp
                 Append n-gram temp to end_punct
             Else
                 Append temp to mid_punct
```

```
                    break
                End if
            End if
        End for
    End for
Return beg_punct, mid_punct, end_punct, whole_punct
End
```

Algorithm 4.2: Punctuation n-grams extraction Pseudocode

Table 4.2: An example of punctuation n-grams for n=3.

	Punctuation n-grams for n=3
Beg_punct	‘.ም.’, ‘. ም’, ‘፣ ጠ’, ‘፡ አ’, ‘፣ ተ’, ‘፡ ቢ’, ‘፣ ታ’, ‘.ም.’, ‘. መ’, ‘፣ ገ’, ‘፡ መ’, ‘፣ ና’, ‘፡ ቢ’, ‘፣ ያ’, ‘፡፡ ‘, ‘፡ አ’, ‘፣ ተ’, ‘፡ አ’, ‘፣ ሲ’, ‘፣ ከ’, ‘፣ ይ’, ‘. ሚ’, ‘፣ ዴ’, ‘፣ በ’, ‘፣ አ’, ‘፣ አ’,
Mid_punct	‘ዓ.ም’, ‘ም. ‘, ‘ት፣ ‘, ‘ት፣ ‘, ‘ዓ.ም’, ‘ም. ‘, ‘ን፣ ‘, ‘ው፣ ‘, ‘ም፣ ‘, ‘ል፣ ‘, ‘ኑ፣ ‘, ‘ት፣ ‘, ‘ም፣ ‘, ‘ባ፣ ‘, ‘ን፣ ‘, ‘1,5’, ‘ም፣ ‘, ‘ጥ፣ ‘, ‘መ፣ ‘, ‘ዓ.ም’, ‘ም. ‘, ‘ት፣ ‘, ‘ት፣ ‘, ‘ዶ/ር’, ‘ን፣ ‘, ‘ን፣ ‘,
End_punct	‘ያት፣’, ‘የቁ፡’, ‘ቁ፡፡’, ‘ቋል፡’, ‘ል፡፡’, ‘ጹት፣’, ‘ዋል፡’, ‘ል፡፡’, ‘ ዓ.’, ‘ቡን፣’, ‘ዋል፡’, ‘ል፡፡’, ‘ጠው፣’, ‘ዋል፡’, ‘ል፡፡’, ‘ቡም፣’, ‘ኗል፡’, ‘ል፡፡’, ‘ታል፡’, ‘ል፡፡’, ‘ዋል፡’, ‘ል፡፡’, ‘ላል፡’, ‘ል፡፡’, ‘ዋል፡’, ‘ል፡፡’,
Whole_punct	‘ዓ.ም’, ‘.ም.’, ‘ም. ‘, ‘. ም’, ‘ያት፣’, ‘ት፣ ‘, ‘፣ ጠ’, ‘የቁ፡’, ‘ቁ፡፡’, ‘፡፡ ‘, ‘፡ አ’,’ቋል፡’, ‘ል፡፡’, ‘ታል፡’, ‘ል፡፡’, ‘፡፡ ‘, ‘፡ አ’, ‘ አበ’, ‘በባ፣’, ‘ባ፣ ‘, ‘፣ ሲ’, ‘ነው፡’, ‘ው፡፡’, ‘፡ ቆ’, ‘ቦት፣’, ‘ት፣ ‘, ‘፣ አ’,

D. Space N-Gram Features

Space n-gram features are extracted as space prefix and space suffix n-grams. The space prefix n-grams will hold n-grams for which the first character is space and the space suffix n-grams will hold n-grams for which the last character is space. The space n-grams are extracted from a normalized text from the normalization component. The Table 4.3 shows examples of space n-grams for n = 2.

Table 4.3: An example of space prefix and space suffix n-grams

	Space n-grams
Space Prefix	‘ል’, ‘ዘ’, ‘ም’, ‘የ’, ‘ጥ’, ‘ጥ’, ‘አ’, ‘የ’, ‘ፉ’, ‘ላ’, ‘ማ’, ‘አ’, ‘ሰ’, ‘ባ’, ‘ጉ’, ‘ወ’, ‘እ’, ‘ስ’, ‘ች’, ‘ተ’, ‘ነ’, ‘የ’, ‘የ’, ‘ም’, ‘በ’, ‘እ’, ‘ስ’, ‘በ’, ‘ሲ’, ‘ቆ’, ‘የ’, ‘ም’, ‘በ’, ‘ብ’, ‘ሳ’, ‘በ’, ‘ረ’, ‘እ’, ‘መ’, ‘በ’,
Space Suffix	‘ጥ ‘, ‘ት ‘, ‘ፍ ‘, ‘ሽ ‘, ‘ቁ ‘, ‘ያ ‘, ‘ጥ ‘, ‘፣ ‘, ‘ጫ ‘, ‘ና ‘, ‘ች ‘, ‘ር ‘, ‘ት ‘, ‘ን ‘, ‘ት ‘, ‘ዔ ‘, ‘ት ‘, ‘ው ‘, ‘ኙ ‘, ‘ች ‘, ‘ው ‘, ‘፡ ‘, ‘ያ ‘, ‘ጥ ‘, ‘ት ‘, ‘ው ‘, ‘ለ ‘, ‘ቱ ‘, ‘ሚ ‘, ‘ር ‘, ‘ቱ ‘, ‘ጥ ‘, ‘ት ‘, ‘ን ‘, ‘ም ‘, ‘፣ ‘, ‘ት ‘, ‘ም ‘, ‘ለ ‘,

E. Part of Speech N-Gram Features

Part of speeches, in a language, are categories of words representing their function within a sentence. Part of speeches are created using a part of speech tagger, a tool which assigns morpho-syntactic information (morphological markings of words subject to their syntactic role) to a token (word) using the contextual information from a sentence depending on the rules of the language in use. In our system, n-gram patterns of these part of speeches (POS n-grams), a set of POS patterns annotated with their respective frequency, are extracted to create author signature.

The part of speech n-grams are used to represent the style of an author by extracting and grouping them into two different categories. The first category holds only part of speech tags and the second category holds word plus part of speech tags. In the first category, which holds only part of speech tags of words, for a sentence an equivalent part of speech representation is used. For example, for the sentence*:”**የኢህአዴግ ምክር ቤት በቀጣዩ ሳምንት ይሰበሰባል።”***, the part of speech tags “***NP N N ADJP N V PUNC***“are used to create the POS n-grams. The second category is created by combining the actual word and its corresponding part of speech tag in a given sentence using an underscore (_). The second category is considered with the assumption that using only occurrences of part of speech tags in a particular author’s writing may not be enough to represent the actual preference of the author on how he/she uses a word in a given sentence. For example, a single word in two different sentences may appear with different syntactic roles or functions. In the sentences: “ህይወት ርግብ የሆነች ሴት ነች።” and “ርግብ የተባለችው ወፍ የየዋህነት ምሳሌ ነች።” the word “ርግብ” appears here in two different sentences with multiple functions. This makes the use of the second category of POS n-grams very much important since it allows to capture which parts of speech tag go with which

word, which couldn't be captured through the use of only part of speeches of words. Algorithm 4.3 shows the extraction of POS n-grams.

```
Algorithm: POS features extraction Pseudocode
Input:  Tagged texts from POS Tagger
Output: words_plus_posTags, posTags
Begin:
            words_plus_posTags, posTags = [ ], [ ]
      For a word w in Tagged texts
            pos_tag = POS tag of w
            append (w + "_" + pos_tag) to Words_plus_posTags
            append pos_tag to posTags
      End for
Return words_plus_posTags, posTags
End
```

Algorithm 4.3: POS features extraction Pseudocode

F. Poem Specific Features

In AA, poems differ from other written texts documents in that when poems are written the poets usually add intentional style preferences which is basically one of the main attributes of a poet in identifying a given an anonymous poem. Other than the different n-gram features we have considered couplets, tercets, quatrains and, end stopped lines (ዝግ ስንኝ) and run on lines (ክፍት ስንኝ), and couplets that rhyme for the eye, couplets that rhyme for the ear and couplets that rhyme for the heart as poem specific features and applied char n-gram extraction on them. We created couplets (two verses), tercets (three verses) and quatrains (four verses) from verses of poems from the versification component in preprocessing. We chose tercets and quatrains as a replacement for stanzas (አረፍተ ስንኝ) as the rules to detect stanzas in Amharic language are not always robust. In Amharic stanzas are detected using the punctuation mark arat netib (::). However, many poets of the language use arat netib not only to mark a stanza but also at the end of a couplet or tercets making it difficult to detect stanzas in a poem robustly. In this study, we applied character n-gram

on these sets of categories as poem specific features. Algorithm 4.4, Algorithm 4.5 and Algorithm 4.6 will devise the extraction of couplets, tercets and quatrains respectively.

Algorithm: Extraction of couplets
Input: Versified poems from versification component
Output: *couplets*
Begin:
create couplets as list
For every poem $p_i \in$ poem_verses, i=1, 2,…,n where n is number of poems in poem_verses
For every verse $v_j \in p_i$, j=1, 2,,m where m is the (number of verses in poem pi) – 1
temp = v_j, v_j+1 # assign two consecutive verses to temp
append temp to temp2 # create list of couplets for poem pi
End for
append temp2 to couplets # create couplets for all poems
End for
Return *couplets*
End

Algorithm 4.4: Extraction of Couplets

Algorithm: Extraction of tercets
Input: Versified poems from versification component
Output: *tercets*
Begin:
create tercets as list
For every poem $p_i \in$ poem_verses, i=1, 2,…,n where n is number of poems in poem_verses

For every verse $v_j \in p_i$, j=1, 2,,m where m is the (number of verses in poem p_i) – 2
temp = v_j, v_j+1, v_j+2 # assign three consecutive verses to temp
append temp to temp2 # create list of tercets for poem pi
End for
append temp2 to tercets # create tercets for all poems
End for
Return *tercets*
End

Algorithm 4.5: Extraction of tercets

Algorithm: Extraction of quatrains
Input: Versified poems from versification component
Output: *quatrains*
Begin:
create quatrains as list
For every poem $p_i \in$ poem_verses, i=1, 2,…,n where n is number of poems in poem_verses
For every verse $v_j \in p_i$, j=1, 2,,m where m is the (number of verses in poem pi) – 3
temp = v_j, v_j+1, v_j+2, v_j+3 # assign four consecutive verses to temp
append temp to temp2 # create list of quatrains for poem pi
End for
append temp2 to quatrains # create quatrains for all poems
End for
Return *quatrains*
End

Algorithm 4.6: Extraction of Quatrains

These algorithms are used to create the categories couplets, tercets and quatrains. Table 4.4 shows examples of the categories for the following poem.

መኖርሽን እንጃ ስሜትሽ ህልም ነው
የራቀው ሰማይ ነሽ የማይዳሰሰው
ተረሳሽው ነሽ ያለፈው ትናንት፤
ማነው የሚያስብሽ እሌለሁበት።።

Table 4.4: Examples of couplets, tercets and quatrains

Couplet	Tercet	Quatrain
መኖርሽን እንጃ ስሜትሽ ህልም ነው የራቀው ሰማይ ነሽ የማይዳሰሰው	መኖርሽን እንጃ ስሜትሽ ህልም ነው የራቀው ሰማይ ነሽ የማይዳሰሰው ተረሳሽው ነሽ ያለፈው ትናንት፤	መኖርሽን እንጃ ስሜትሽ ህልም ነው የራቀው ሰማይ ነሽ የማይዳሰሰው ተረሳሽው ነሽ ያለፈው ትናንት፤ ማነው የሚያስብሽ እሌለሁበት።።

The other categories we considered as a poem specific features are end stopped lines and run on line verses. In poems, end stopped lines are verses that end with a punctuation and run on lines are verses in which their end is not a punctuation. Algorithm 4.7 shows the extraction of end stopped line and run on line verses.

Algorithm: Extraction of end stopped and run on line verses

Input: Versified poems from versification component

Output: *end_stopped and run on line verses*

Begin:

```
create end_stopped_lines, run_on_lines as list
For every poem pi ∈ poem_verses, i=1, 2,…,n where n is number of poems in poem_verses
    For every verse vj ∈ pi, j=1, 2,,m where m is the (number of verses in poem pi)
        temp = word_punct_tokenize(vj)  # tokenize verse vj into words and punctuations and assign to temp
        temp2 = temp[-1] # assign last element in temp to temp2
    If temp2 is an Amharic punctuation:
        append verse vj to temp3 # create list of end stopped line verses
    Else:
        append verse vj to temp4 # create list of run on line verses
    End for
    append temp3 to end_stopped_lines
    append temp4 to run_on_lines
End for
```

Return *end_stopped_lines, run_on_lines*

End

Algorithm 4.7: Extraction of end stopped and run on line verses

Table 4.5 shows examples of end stopped line and run on line verses for the poem above.

Table 4.5: End stopped line and Run on line verses

End stopped line verses	Run on line verses
ተረሳሽው ነሽ ያለፈው ትናንት፤ ማነው የሚያስብሽ እሌለሁበት።	መኖርሽን እንጃ ስሜትሽ ህልም ነው የራቀው ሰማይ ነሽ የማይዳሰሰው

Rhyming is a primary characteristics of a poem in which poets use to produce sensible sounds to readers. Poets follow different rhyming techniques and there exist different rhyming techniques in each type of language. In Amharic, types of rhymes involves rhymes for the eye (ለዓይን), rhymes for the ear (ለጆሮ) and rhymes of the heart (ለልብ). Rhymes for the eye involves verses in which their rhyming letter is identical. Rhymes for the ear are verses that rhyme with identical vowels not necessarily letters. And the last type of rhyming, rhymes for the heart, which includes verses that doesn't possess both characteristics of rhymed for the eye and the ear. In this work, we categorized couplets into these three categories. Algorithm 4.8 devises the process.

```
Algorithm: Categorize rhyming couplets into categories: for the eye, for the ear and for the heart
Input:  Couplets
Output: Couplet categories
Begin:
    create for_the_eye, for_the_ear, for_the_heart as list
    For every couplet ci ∈ couplets, i=1, 2,…,n where n is number of couplet in couplets
            verse1 = couplet[0] # assign verse 1 of couplet ci to temp1
            verse2 = couplet[1] # assign verse 1 of couplet ci to temp2
            temp1 = word_punct_tokenize(verse1)  # tokenize verse1 into words and punctuations and assign to temp3
            temp2 = word_punct_tokenize(verse2)
            temp3 = temp1.remove_punct  # Remove Amharic punctuations from temp1
            temp4 = temp2.remove_punct
            last_word_verse1 = temp3[-1] # assign last element of temp3 to last_word_verse1
            last_word_verse2 = temp4[-1]
            temp5 = last_word_verse1.transliterate() # transliterate last word in verse 1 into Latin alphabets and assign to temp5
            temp6 = last_word_verse2.transliterate
            If last_word_verse1[-1] equals last_word_verse2[-1]:
                    append couplet ci to for_the_eye # if last characters are similar
             Else if temp5[-1] equals temp6[-1]:
```

```
                append couplet ci to for_the_ear # if last vowels are
            similar
            Else:
                append couplet ci to for_the_heart # if both last
            characters and vowels are not similar
        End for
Return for_the_eye, for_the_ear, for_the_heart
End
```

Algorithm 4.8: Categorize rhyming couplets into categories: for the eye, for the ear and for the heart

4.2.4 Feature Concatenation

Feature concatenation is the process of combining different sets of features from multiple feature extraction components. The union of multiple (different) features is expected to bring the best out of a classifier in a classification system improving the accuracy to some extent.

In this study, the feature concatenation component is used to create combinations of the different sets of feature extraction methods considered in this study to extract the different set of features (word n-gram, char n-gram, punctuation n-gram, space n-gram and POS n-gram features). In creating the combined feature set, instead of combining the features after extraction, the feature extraction methods used to extract the individual n-gram features are combined creating a single transformer that can be used to produce concatenated features as required. For example, the methods used to create char n-gram and word n-gram features can be combined to create a single feature set of char and word n-grams at once. The output of the feature concatenation component, concatenated feature sets, is directly fed to dimensionality reduction component for dimensionality reduction.

4.2.5 Dimension Reduction

Dimension reduction is the process of converting a given data with bigger dimensions into smaller dimensions yet still manage to keep same information. This task is performed just before training the classifier so all the redundant and irrelevant data from the large features could be removed. The feature extraction stage (using both tf and tfidf weighting) transforms the data into sparse vectors with a very large dimensions. The feature concatenation component then creates a union of sparse vectors for the different type of features considered producing higher dimensions.

The input to the dimension reduction component is sparse matrices from the feature concatenation component. To handle the higher dimensionality of the sparse vectors produced an extension of the principal component analysis dimension technique, the sparse principal component analysis (SPCA) is used. SPCA finds a linear combination of the high dimensional sparse vectors containing very few of them. Table 4.6 shows an example of how SPCA reduces dimensionality of sparse vectors of the combination of character and POS n-gram features for the poem dataset.

Table 4.6: An example of SPCA for poem dataset using char and POS n-grams

Input: Sparse matrices of the union of character and POS n-grams higher dimension	**Output:** A transformed array with reduced dimensions
Shape/ Dimension: (198, 1000)	**Shape/ Dimension:** (198, 8)
Sparse Matrix: Assuming columns as (A,B) C (A: documents index, B: specific word-vector index) C: TF Score for word B in Document A	
(0, 9) 11 (0, 11) 2 (0, 12) 3 (0, 15) 5 ..... (197, 995) 1 (197, 996) 17 (197, 997) 1 (197, 998) 1 (197, 999) 4	[[13.8953865 36.95493249 8.2412921 ... 12.10657536 -4.27043528 6.15623495] [18.25657268 35.46457084 9.96912041 ... 11.48372124 -3.31276102 7.92833925] ..... [21.5845552 -170.42260402 12.43545594 ... -25.77778036 -3.13597989 -13.10210718] [24.5455318 -181.1199781 9.52057993 ... -3.36217838 -23.98528528 19.09942406]]

4.2.6 Classifier Training

When it comes to classifiers, there are two things to consider: training and classification. Training the classifier involves to use samples of predefined classes or sets of nodes (training sets) to train the classifier with what is inside those classes. In this study, classes of an author's writing style representation are created with each class labeled with the author's name so it can be used to learn the classifier selected for the attribution process. A support vector machine (SVM) classifier is trained with all the n-gram features extracted and built as an author profile. The author profiles

consists of feature vectors with the name of the author being used as class names. These class names used are provided under the Annex section. The output of this component is an author learning model created by training an SVM classifier with the features extracted.

4.2.7 Author Learning Model

In the problem of authorship analysis it is required to create a representation of an author's writing style so it could be possible for an anonymous text to be identified. The different types of n-gram features: word n-gram features, character n-gram features of all forms, punctuation n-gram features, space n-gram features and POS n-gram features for each article of an author's writings are used to create the individual author profile. For authors of poems, poem specific features along with other features are considered as the poet's profile. Thus, an SVM classifier is trained with all the extracted features to build the author learning model. This author model will be used to predict the name of an anonymous text in the classification process.

4.2.8 Classification/ Authorship Attribution

Author attribution (Authorship Identification) involves identifying the names of authors of anonymous texts. The author identification task is done using the SVM classifier which predicts the author of a given anonymous text using the different types of n-gram features from which the classifier is trained on. In predicting the author, the SVM classifier compares style features extracted from the test document with each candidate author's profile and returns the name of the specific author from a closed set of candidate authors for which the test document belongs to. The prediction is done by the support vector machine's classifier support vector classifier (SVC). Since the problem is a closed set attribution problem, the classifier considers the authors as class names and the concept of support vectors is used to classify the test document to the classes. SVC applies one vs one approach for multi-class classification task. As per the concept of the one vs one approach, n*(n-1)/2 binary classifiers will be built, where n is the number of classes, which is one classifier per pair of classes. In making predictions, all the classifiers built make votes and the class (name of the author in this case) with the most votes (highest confidence score) by the classifiers is selected as the predicted class (Author) name.

Chapter Five: Experiment

5.1 Introduction

Experimentation is performed to test and justify the proposed attribution model for the Amharic language. The chapter is divided into five sections: Section 5.2 describes the experimental setup (tools, programming language, packages, experimental settings and prototype of the model) used to carry out the experiment. Section 5.3 shows the experimental procedures followed to perform experiment with results. Under Section 5.4, evaluation of the results is shown and the last section of the chapter, Section 5.5, presents the overall discussions.

5.2 Experimental Setup

5.2.1 Data Collection

The proposed model is experimented using two datasets. The first dataset is built by collecting text data from *Kumneger magazine* and *The Reporter Ethiopia Media & Communications Center*. It comprises sets of article writings of authors who are journalists of news articles and stories on newspapers and magazines. These writings (articles) are written between the years 2013 to 2019. More than 2000 articles of 20 authors are used to create the dataset. The second dataset is built using poems. These poems are written by poets: Bewketu Seyoum and Gebrekristos Desta. We decided to create this dataset for the language because for authorship analysis studies, such datasets are more topic independent since the articles are written on different topics and when writing different articles on different topics authors tend to introduce new writing styles giving the model being developed an improved performance. The first dataset constitutes nine topics written by the twenty authors and the second dataset contains more than 120 poems of two poets. Table 5.1 shows the dataset statistics:

Table 5.1: Dataset Statistics

Dataset	Authors	Articles	Topics	Sentences	Words
The Reporter Ethiopia Media & Communications Center	18	1904	Social, Sport, Politics, Business, Kinina Bahil (ኪነ ባህል), World, Sport, Creation, Zinik (ዝንቅ)	68,300	> 1.2 million
Kumneger Magazine	2	166	Politics, Business, Social, Sport	10,062	156,566

Table 5.1: Poem Dataset Statistics

Dataset	Poets	Poems	Verses	Words
Poem	Bewketu Seyoum	63	1392	5120
	Gebrekristos Desta	40	1966	6499

5.2.3 Packages, Tools and Programming Language

In developing a prototype and experimenting the authorship attribution model, different types of fitting packages/tools and programming languages are carefully chosen. The different types of algorithms to extract the different sets of n-gram features (char n-grams, word n-grams, punctuation n-grams, space n-grams, POS n-grams, Poem specific features) defined in the previous chapter are implemented using the python[1] programming language. Python, a high level general purpose programming language, is preferred for its simplicity in code readability and support for multiple programing paradigms. Python, version 3.4, is used to develop the prototype of our model.

5.2.3 Experimental Settings

The prototype of the authorship attribution model is developed and tested using Toshiba satellite laptop. Experiments are all done on the same laptop with:

- Windows 10 64-bit operating system,

- 6 GB of RAM and 740 GB of SDD,
- Processor – Intel Core i5-4200U, CPU @ 1.60GHz, 2301 Mhz

The performance of the model will substantially be enhanced with a much better hardware since large author dataset, maximized use of the different feature types and extraction of larger frequent words set requires much more hardware resources.

5.2.4 Prototype of the model

The prototype is provided with a user interface that will allow a user to import an article or a poem as a text (.txt) file to the system and predict the author of the provided text file. The interface shows the functionalities of the model, which involves importing a file and analyzing the file so the author of the given text file is predicted. The Figures 5.1 to 5.3 show the user interface through the different functions.

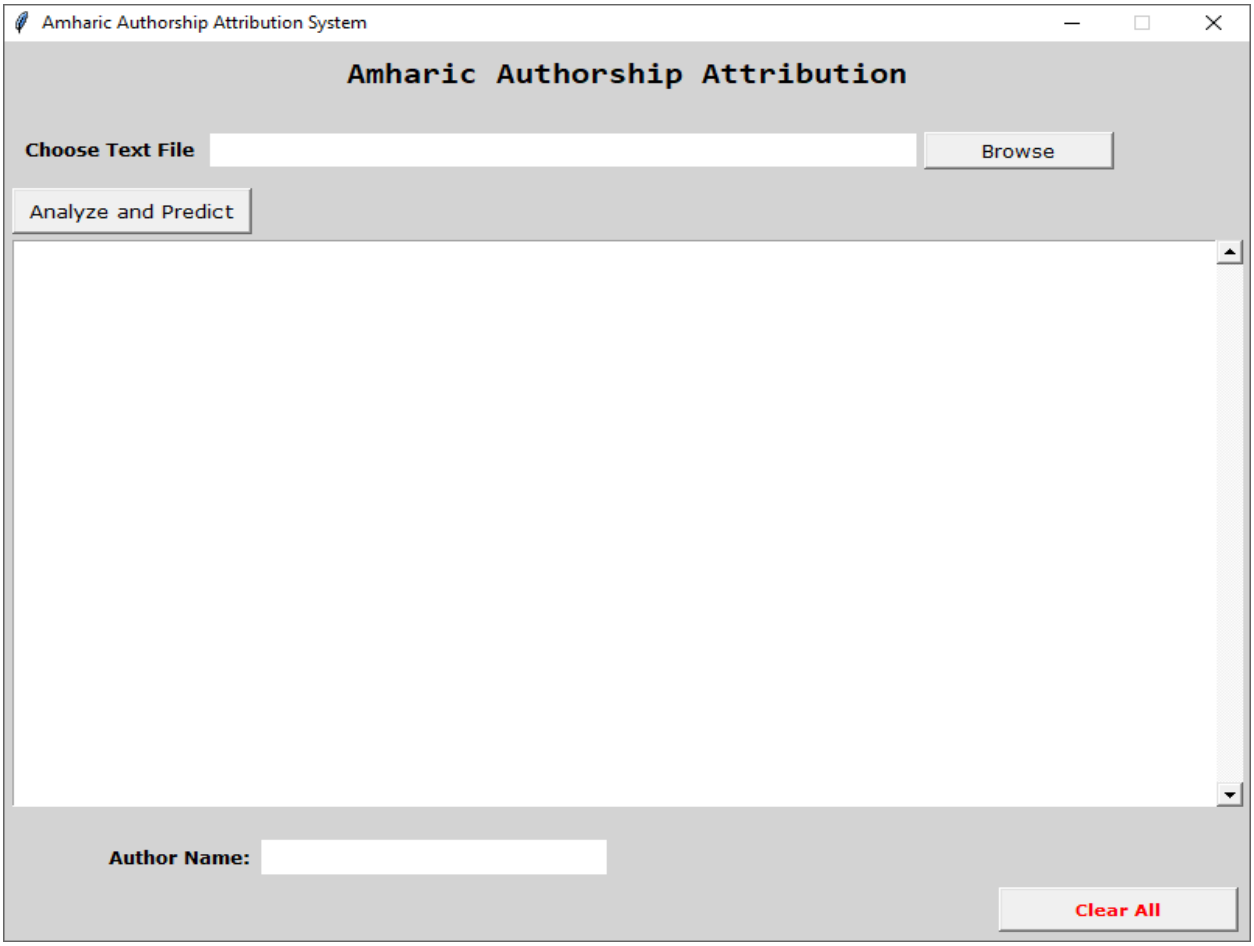

Figure 5.1: System User Interface

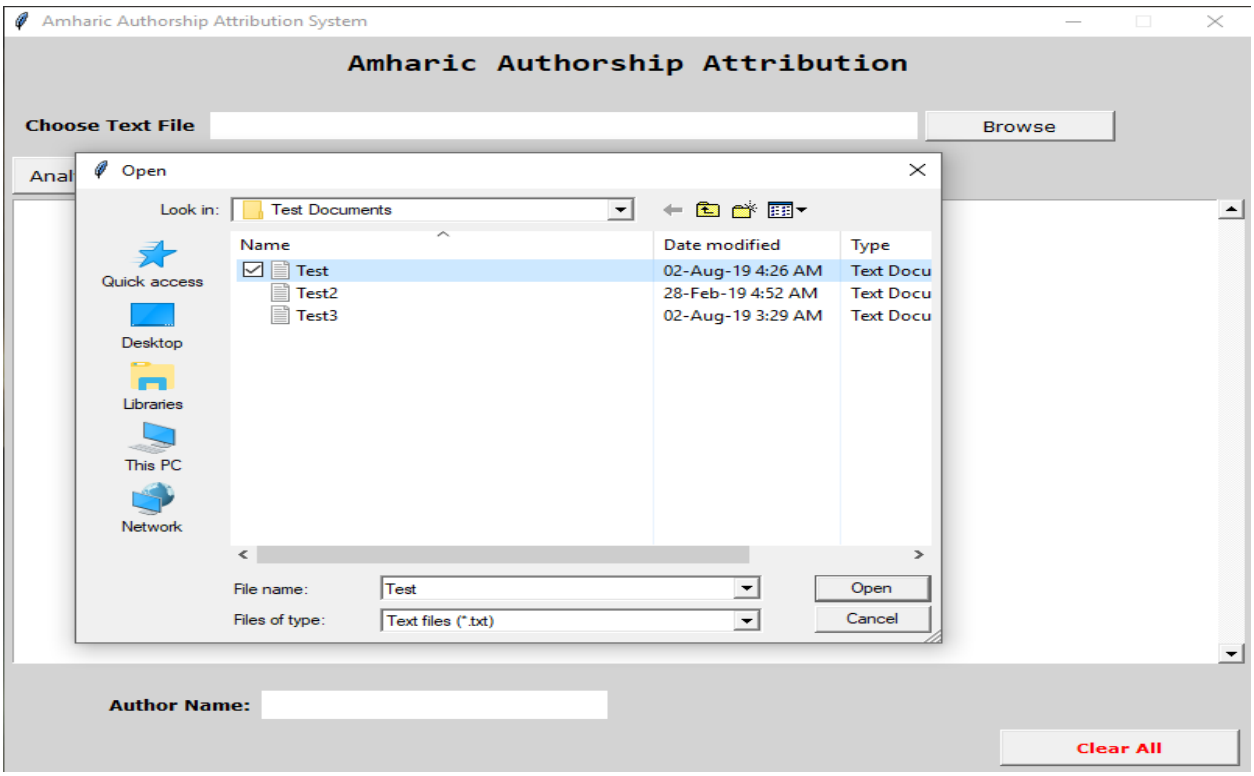

Figure 5.2: Import new test file

Figure 5.3: Attribution Result User Interface

5.3 Experiment

In this section, to test the effectiveness of the proposed model, we have performed different experiments using the different categories of n-gram features extracted. Each category of the n-gram feature is tested to identify which are the most author discriminators amongst them. We have also tested the different combinations of each category of n-grams for both datasets. For poems, poem specific features are extracted and used for experimenting. To transliterate Amharic words to Latin alphabets we used the SERA transliteration scheme [110]. For testing, in order to find the appropriate tuning of the parameter n, different values of n are chosen for each category of n-gram feature. These n-gram features are extracted and weighed using term frequency (tf) and term frequency inverse document frequency (tfidf) feature extraction methods. To perform term frequency and term frequency inverse document frequency, we have used the python Sklearn's feature extraction classes (transformer objects) *CountVectorizer* and *TfidfVecorizer* respectively.

To create the different combinations of the features we used the python Sklearn's feature decomposition class, *FeatureUnion*. The *FeatureUnion* class combines results of multiple transformer objects into a new transformer that combines their output. In the system, results from the *CountVectorizer* and *TfidfVectorizer* transformers for each type of n-gram features are combined. Both transformer objects produce sparse matrices of the different n-gram features. The combination of multiple transformer objects will significantly increase the shape of the sparse matrices. To handle this increase in dimensionality, the python Sklearn's decomposition class *SparsePCA*, an alternative to PCA (Principal component analysis), which is capable of functioning with sparse matrices is used.

On the other hand, one of the most important factors in the attribution process, the value k, which decides how much of the least frequent words of the language could be used to create a distorted form of the original text is chosen by carrying out rigorous testing for k = 0, 100, 200, 500, 1000, 4000 and 5000. We did not go for k values more than 5000 for it is against the limitation of the available resource we had for testing our model. The distorted texts are used for word n-gram and char n-gram features. And also, for choosing the length of n in experimenting the different types of n-gram features, we have used various n values, which are most common n values in many authorship attribution studies, as shown in Table 5.2:

Table 5.2: Value of n for each n-gram feature tested

N-gram Feature	n values
Word n-gram	2,3,4,5,6
Char n-gram	3,4,5,6,7
POS n-gram	3,4,5
Punctuation n-gram	3,4,5
Space n-gram	2,3,4
Poem specific n-gram	3,4,5

Classification is done using the python Sklearn's SVM (support vector machine) classifier SVC (support vector classifier). The parameters of this SVC classifier: kernel type ('linear', 'rbf'), gamma (Kernel coefficient for 'rbf'), and C (Penalty parameter of the error term) are estimated using the Grid Search, an approach that allows tuning of parameter values to build and evaluate a model. After tuning the parameters for the classifier an SVM classifier with a linear kernel is built.

The model is also tested using the Naive Bayes algorithm. All the n-gram based features considered in our model are tested using the Multinomial Naive Bayes method which is one of the many methods of Naive Bayes algorithm. We performed comparisons between the SVM and Naive Bayes for the best features, and presented the prediction performances of each of the classification algorithms.

Generally, we have held experiments for all categories of n-gram features extracted for representing the style of each author's writing. We have also created and experimented different combination of these n-gram features which recorded the best accuracy by themselves. The author dataset is divided into training and test sets. Demanding the avoidance of the variance problem which usually happens when one splits a dataset into training and test set, we have held a k-fold cross validation for k value of 10.

5.4 Results and Evaluation

5.4.1 Test Results

The model is tested for all categories of features for different values of n and least frequent words set k using a 10-fold cross validation method using both the datasets. Features weighted using both term frequency (tf) and term frequency inverse document frequency (tfidf) are considered. In

testing character and word n-grams we also chose longer n-grams (n = 6, 7) so as to see how the text distortion technique can contribute to creating a topic independent model since longer n-grams are able to capture topic information. The results of the best performances of the classifier using the first dataset for each category of n-gram features are shown in Tables 5.3 – 5.8 for different values of n (n-gram length) and k (number of least frequent words). All these results shown through the tables are discussed under section 5.5, Discussion, of this chapter.

Table 5.3: Accuracy results for character and word n-grams for different values of k and n using both tf and tfidf for both SVM and Naïve Bayes Algorithms

			SVM		**Naive Bayes**	
	n	**k**	**tf**	**tfidf**	**tf**	**tfidf**
Char n-grams	3	**0**	**0.80**	**0.45**	**0.76**	**0.67**
		1000	0.78	0.44	0.76	0.66
		4000	0.78	0.43	0.75	0.67
		5000	**0.80**	0.40	0.75	0.66
	4	**0**	**0.80**	**0.51**	**0.78**	**0.70**
		1000	0.79	0.48	0.78	0.70
		4000	0.79	0.47	0.77	0.69
		5000	0.78	0.47	0.77	0.70
	5	**0**	**0.80**	**0.49**	**0.80**	**0.72**
		1000	0.78	0.48	0.79	0.72
		4000	0.78	0.47	0.80	0.71
		5000	0.77	0.47	0.79	0.71
	6	**0**	**0.80**	**0.47**	**0.80**	0.72
		1000	0.78	0.47	0.80	0.72
		4000	0.78	0.46	0.79	**0.73**
		5000	0.77	0.46	0.80	0.73
	7	**0**	**0.76**	**0.73**	**0.80**	**0.73**
		1000	**0.76**	**0.72**	0.80	0.74
		4000	**0.78**	**0.73**	0.79	0.73
		5000	**0.76**	**0.71**	0.80	0.72
Word n-grams	2	**0**	**0.75**	**0.40**	**0.78**	**0.77**
		1000	0.74	0.32	0.78	0.77
		4000	0.73	0.32	0.78	0.76
		5000	0.73	0.25	0.77	0.75
	3	**0**	**0.63**	0.17	**0.63**	**0.65**
		1000	0.56	**0.26**	0.62	0.64
		4000	0.57	0.18	0.63	0.64
		5000	0.57	0.15	0.63	0.64
	4	**0**	**0.42**	**0.17**	**0.39**	**0.39**
		1000	0.30	0.12	0.36	0.38
		4000	0.23	0.11	0.35	0.37
		5000	0.30	0.10	0.37	0.39
	5	**0**	**0.34**	**0.08**	**0.20**	**0.21**
		1000	0.17	0.07	0.21	0.20
		4000	0.16	0.07	0.19	0.20
		5000	0.16	0.07	0.20	0.20
	6	**0**	**0.10**	**0.39**	**0.13**	0.12
		1000	0.10	0.39	0.12	0.11
		4000	0.10	0.39	0.12	**0.13**
		5000	0.09	0.39	0.12	0.11

Table 5.4: Accuracy results for Sentence length character n-grams using both tf and tfidf

		SVM		Naïve Bayes	
n-gram type	**n**	**tf**	**tfidf**	**tf**	**tfidf**
SL char n-grams	3	0.39	0.39	**0.46**	**0.47**
	4	**0.40**	**0..37**	0.46	0.46
	5	0.40	0.36	0.46	0.46

Table 5.5: Accuracy results for punctuation n-grams using both tf and tfidf

		SVM		Naïve Bayes	
n-gram type	**n**	**tf**	**tfidf**	**tf**	**tfidf**
Whole punctuation	3	0.70	0.23	**0.69**	0.57
	4	**0.71**	0.30	0.69	0.59
	5	0.70	**0.44**	0.67	**0.60**
Beg punctuation	3	0.62	0.20	0.60	0.43
	4	**0.65**	0.36	**0.68**	0.56
	5	0.64	**0.42**	0.66	**0.59**
Mid punctuation	3	0.50	0.19	0.48	0.41
	4	0.60	0.30	0.61	0.53
	5	**0.67**	**0.36**	**0.66**	**0.56**
End punctuation	3	0.62	0.23	**0.63**	0.45
	4	0.63	0.30	0.63	0.54
	5	**0.64**	**0.36**	0.63	**0.55**

Table 5.6: Accuracy results for space n-grams using both tf and tfidf

		SVM		Naïve Bayes	
n-gram type	**n**	**tf**	**tfidf**	**tf**	**tfidf**
Space prefix	2	0.62	0.06	0.61	0.37
	3	0.74	0.36	0.75	0.64
	4	**0.77**	0.45	**0.77**	**0.69**
	5	0.76	**0.46**	0.77	0.69
Space suffix	2	0.62	0.08	0.63	0.32
	3	0.74	0.45	0.76	0.67
	4	0.76	**0.50**	**0.78**	**0.69**
	5	**0.77**	0.49	0.78	0.71

Table 5.7: Accuracy results of part of speech n-grams using both tf and tfidf

		SVM		Naïve Bayes	
n-gram type	**n**	**tf**	**tfidf**	**tf**	**tfidf**
POS n-grams	3	0.55	0.06	0.40	0.18
	4	0.59	0.10	0.50	0.19
	5	0.60	0.13	0.55	0.22
	6	**0.61**	**0.15**	**0.60**	**0.25**

Words_plus_POS n-grams	3	0.77	0.09	0.76	0.53
	4	**0.78**	0.20	0.76	0.59
	5	0.78	0.33	**0.77**	0.62
	6	0.78	**0.40**	0.77	**0.64**

Table 5.8: Accuracy results of char and word n-grams using both tf and tfidf for stemmed texts

		SVM		**Naïve Bayes**	
n-gram type	**n**	**tf**	**tfidf**	**tf**	**tfidf**
Char n-grams	3	0.78	0.24	**0.80**	0.71
	4	0.76	0.23	**0.78**	0.72
	5	**0.78**	0.18	0.76	0.71
Word n-grams	3	0.56	0.07	**0.70**	0.64
	4	0.37	0.06	0.52	0.55
	5	0.17	0.06	0.35	0.36

Apart from the different sets of n-gram features extracted and tested, we have created different combination of these n-gram features and presented the union of features that has achieved the best accuracy.

Table 5.9: Accuracy results for best n-gram combinations using term frequency

	SVM	**Naïve Bayes**
n-gram union	**Tf**	**Tf**
Char 3-gram and Space suffix 5-gram	0.83	**0.67**
Char 3-gram and Space suffix 4-gram	0.82	0.65
Char 3-gram and Words plus POS 4-gram	**0.87**	0.25
Char 4-gram and Words plus POS 4-gram	0.84	0.32
Char 5-gram and Words plus POS 4-gram	0.84	0.35

The results of the best performances of the classifier using the poem dataset for each category of poem specific char n-gram features are shown in Tables 5.10 for different values of n (n-gram length).

Table 5.10: Accuracy results for poem specific features using tf for both SVM and Naive Bayes

	SVM	**Naïve Bayes**
n-gram type	**tf**	**tf**
Char n-gram on whole poem	0.90	0.95
Char n-gram on stemmed poems	0.90	0.85

Pos n-gram on whole poem	Pos	0.76	0.76
	Word_plus_pos	0.90	0.95
Punctuation n-gram on whole poem	Beg Punct	0.92	0.95
	Mid Punct	0.90	0.81
	End Punct	0.94	0.90
	Whole Punct	0.94	0.95
Space n-gram on whole poem	Space prefix	0.81	0.90
	Space suffix	0.95	0.91
End stopped lines		0.96	0.90
Run on lines		0.90	0.88
Couplets, Tercets, Quatrains		0.95	0.87
Rhymed for the eye		0.92	0.95
Rhymed for the ear		0.81	0.85
Rhymed for the heart		0.92	0.91

5.4.2 Evaluation Results

The model is evaluated using the various and most common evaluation metrics (precision, recall, and f-score). Precision shows how relevant the result is, meaning it indicates how many times the classifier is correct in attributing a given text. Recall states how many texts from a set of predictions are correctly attributed to the rightful author by the classifier. F-score is the harmonic mean of precision and recall. The authorship attribution task is done for a closed set authorship attribution problem. The performance of the proposed model is evaluated using the above mentioned metrics for the n-gram feature which achieved the best accuracy. Table 5.9 shows the average scores for precision, recall and f-score.

Table 5.11: Evaluation Result for both SVM and Naïve Bayes Classifiers for dataset 1

Classifier	No of classes	N-gram Feature	Training Time	Prediction Time	Accuracy	Precision	Recall	F-score
SVM	20	combination of character 3-grams and words_plus_POS 4-grams	413.088 s	32.909 s	0.87	0.88	0.87	0.87
Naive Bayes	20	character 5-grams	0.162 s	0.036 s	0.80	0.67	0.67	0.67

Table 5.12: Evaluation Result for both SVM and Naïve Bayes Classifiers for dataset 2

Classifier	No of classes	N-gram Feature	Training Time	Prediction Time	Accuracy	Precision	Recall	F-score
SVM	2	End Stopped Lines	0.118 s	0.019 s	0.96	0.97	0.93	0.94
Naive Bayes	2	Char n-gram on whole poem	0.001 s	0.013 s	0.95	0.96	0.94	0.94

A confusion matrix is used to visualize the summary of overall classification performance of the attribution system for a SVM classifier. In confusion matrix, the diagonal elements are used to show the number of correct predictions made by the classifier used and those elements outside the diagonal are wrongly predicted by the classifier. The confusion matrix in Figure 5.4 shows the classifier made utmost predictions correctly and misclassified very few of them.

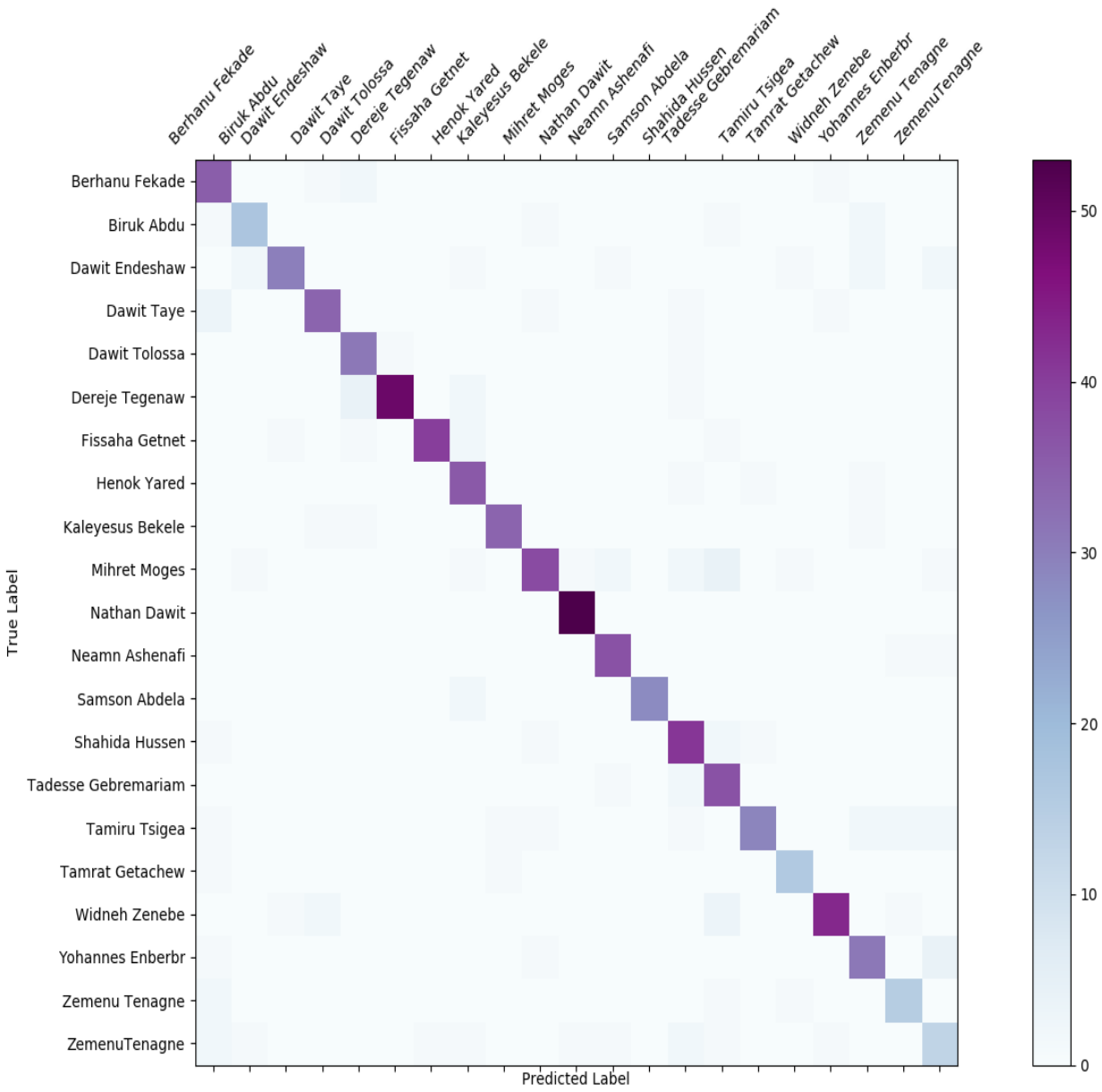

Figure 5.4: Confusion Matrix for SVM classifier for dataset 1

5.5 Discussion

As it is all shown in the test results section of this chapter, the model has achieved varying accuracy for different features using both SVM and Naïve Bayes algorithms for the two datasets. In this section, we will discuss the different test results presented in the tables above for each category of n-gram feature.

The test results for character and word n-grams in Table 5.3 show that the model using the SVM algorithm, except for word 5 and 6 grams, achieved the best results using **tf** instead of **tfidf**. The results also show that the text distortion technique considered doesn't have much of an effect in degrading the accuracy reducing topic dependent ability of the model. Compared to word n-grams, character n-grams achieved better results for most of the tests. The Naïve Bayes algorithm performed very similarly for most of the word and char n-grams using **tf**. However, the Naïve Bayes algorithm, compared to the SVM, performed very well using **tfidf** for char 5, 6 and 7 grams, and word bigrams. This shows the Naïve Bayes algorithm is a good choice to make attribution using tfidf based features.

The sentence length character n-grams, Table 5.4, achieved somewhat the least accuracies compared to other n-gram features for both tf and tfidf using both SVM and Naïve Bayes algorithms. We believe this is because of the boundary of the sentence length categories defined may have resulted in poor performance. Redefining the sentence length boundary or applying different techniques to the existing categories may prove significant.

In punctuation based character n-grams, Table 5.5, **tf** based features still performed better than tfidf features for all values of n tested. For these tf based punctuation n-gram features, both the Naïve Bayes and SVM algorithms achieved somewhat similar results. From the four types of punctuation based character n-grams, whole punctuation n-grams achieved the better accuracies for all three n values tested. The other three types of punctuation n-grams, which performed very much similarly, also achieved significant results. The tfidf based features also showed promising results using the Naïve Bayes algorithm. These shows that Amharic punctuations can be very useful in capturing an author's style of writing.

Space n-gram features, Table 5.6, which are not the most common type of n-gram features, performed better than expected for different n values. Both space prefix and space suffix n-grams,

using SVM and Naïve Bayes algorithms, achieved better results and can be used for Amharic authorship attribution.

Part of speech n-grams, Table 5.7, which are, just like punctuation n-grams, capable of capturing syntactic features of an author's writing are tested and achieved one of the best results for word_plus_pos n-grams. Although POS tags performed well, word_plus_pos tags performed better for both tf and tfidf based features. Same as the other n-gram features the Naïve Bayes algorithm performed better for the tfidf based features.

Character and word n-gram features extracted from stemmed texts of are tested for both tf and tf-idf for SVM and Naïve Bayes algorithms. Table 5.8 shows results of the experiment for n values 3, 4 and 6. The model achieved the highest accuracy for the Naïve Bayes algorithm using char 3 grams and the lowest for word n-grams using tf-idf weighting. The SVM algorithm also performed better with character n-grams using tf weighting.

The last type of features tested using the first dataset which is created by combining the different type of n-gram features tested achieved the biggest accuracy of all using the SVM algorithm as indicated in Table 5.9. The union of Char 3-gram and Words plus POS 4-gram has achieved the maximum accuracy for tf based features. However, the Naïve Bayes algorithm failed to achieve better results for all these different combination of n-gram features. Even for the tfidf based features for which it performed better than the SVM algorithm, the Naïve Bayes algorithm failed to achieve good results for the different combinations of them.

The poem specific features, Table 5.10, are tested using the poem dataset for both SVM and Naïve Bayes classifiers using tf. The model achieved the highest accuracy score of all for the poem specific features using the poem dataset. This shows that, even though the number of classes (poets) in the poem dataset is small the poem features considered prove to be discriminative and are capable of capturing a poet's style of writing poems in the language. The Naïve Bayes performed slightly better than the SVM for this features.

Generally, the overall performance of the model shows that the problem of authorship attribution can be tackled in the Amharic language. The model only misclassified very few classes. Although tfidf based features using the Naïve Bayes algorithm achieved promising results, the results for most of the tests imply that term frequency weighted features are significantly suited for authorship

attribution studies than term frequency inverse document frequency features. Character n-gram, whole punctuation n-gram, space n-gram and word plus POS n-gram features achieved more than 70% accuracy for different values of n. With the exception of word bigrams, word n-gram features tend to be less distinguishing features next to sentence length n-gram features.

Chapter Six: Conclusion and Future Works

In this chapter, the conclusion which gives the details of the general activities done in this study including findings, the contributions of this research work and future works that are beyond the scope of this study are presented in three sections.

6.1 Conclusion

Nowadays, the availability of huge amount of text documents produced anonymously by various individuals is growing rapidly. The anonymity of these text documents is causing bigger problems in forensic applications and when these texts are spam emails, offensive online contents or important documents where multiple individuals are in dispute over its ownership, the problem becomes even bigger. Authorship analysis studies try to address this issue by analyzing several of an individual's writings. Authorship attribution is one of authorship analysis studies which assigns a given anonymous text to a given candidate author.

This research work was proposed to address the problem of authorship attribution in the Amharic language. In order to carry on this study and understand the problem better, we revised the fundamental approaches (statistical, machine learning and hybrid) followed, the different style based features used and classification techniques employed in several research works in the field of authorship analysis. We investigated some topic based Amharic text classification studies to achieve an understanding of the linguistic properties of the Amharic language. We have also reviewed some related authorship attribution works depending on the language they are developed for. With the understanding of the problem area and the language characteristics, the proposed attribution model is developed. The architecture of the model has two main phases: training and attribution. The training phase comprise preprocessing, feature extraction, feature concatenation, dimension reduction and classifier training components. The attribution phase consists all the components in the training phase except the classifier training component but includes another component, the author attribution component.

In developing the attribution model, several tools and attribution methods are considered and used in the study. In preprocessing, a text distortion technique is adopted and a part of speech tagger tool is used to create part of speeches of words. Different type of features: lexical (through word n-grams and sentence length features), character (through character n-grams), syntactic (through

punctuation n-grams and part of speech n-grams) and n-gram based poem specific features are used to represent the style of an author's writing. A feature concatenation technique is used create combinations of the different sets of features. Based on these features, a classifier makes decision on the ownership of a given anonymous text. Finally, to verify the proposed attribution model, experimentation is carried out using Python programming language. A corpus of more than 2000 articles of 20 authors and more than 120 poems of 2 poets is developed and used for the experiment. A support vector machine (SVM) classifier and a Naïve Bayes classifier are used for testing the proposed model. Both of these classifiers obtained different results for the different sets of features considered including the unusual types of features in AA, word_plus_pos tag features, in this study. For tf-idf based features the model achieved better results using Naïve Bayes classifier compared to SVM. For tf based features the model achieved better or similar results using SVM classifier compared to Naïve Bayes. Generally, the model achieved the best accuracy of 86.77 % for the combination of tf based character 3-grams and words_plus_POS 4-grams with an average precision being 88 %, recall 87 % and F-score 87 % using SVM classifier. The findings of this study are promising and show that further studies in the language, given its complexity, can provide improved performance.

6.2 Contributions of the study

The findings of this research work are:

- Studied the problem of the authorship attribution in the field of authorship analysis for the Amharic language which will help open the door for the study of other problems in the field for the language.
- Developed a corpus suitable, not only for authorship attribution, but also, for the various authorship analysis studies (such as Authorship Verification and Author Profiling) in the language.
- Developed an algorithm to apply sentence length features in authorship attribution in a new way through character n-grams.
- Applied word_plus_pos tag n-gram features to the authorship attribution problem and found the combination of word_plus_pos 4-grams and char tri-grams as an effective style markers.
- Studied the attribution problem in Amharic poems and, identified and applied poem specific features in the language.

6.3 Future works

The core point of this study was to develop a model that is capable of attributing a given anonymous text to the rightful author from a set of possible candidate authors. However, such closed set attribution systems may limit the possible owner of a given document to only a set of candidate authors. Apart from increasing the candidate authors set size, open set attribution is one way to tackle this problem. Even though our model achieved significant results, its performance can still be improved. Future work will hold the following tasks:

- Constructing a corpus which is larger than the currently developed one by increasing both the number of authors and the number of documents per author.
- Apply advanced NLP techniques (morphological analysis and syntactic parser) for the language to make use of complex syntactic features like affix n-grams and syntactic patterns.
- Study the proposed sentence length features by redefining the length of sentences per categories to enhance system performance.
- Apply sentence length features using the Chi-square method for the task of authorship verification.
- Study syllabification of poems in the Amharic language to make use of meters and rhythms, initial and internal rhymes.
- Perform further studies on other authorship analysis studies in the language like:
    - Open set authorship attribution
    - Author profiling
    - Authorship attribution of multi-authored texts/documents
    - Genre Classification

References

[1] A. F. Otoom, E. E. Abdallah, M. Hammad, M. Bsoul and A. E. Abdallah, "An intelligent system for author attribution based on a hybrid feature set," *Int. J. Advanced Intelligence Paradigms,* vol. 6, no. 4, 2014.

[2] F. Mosteller and D. L. Wallace, "Inference and Disputed Authorship: The Federalist," *the Journal of Interdisciplinary History,* vol. 1, no. 3, pp. 557-560, 1971.

[3] E. Stamatatos, "A Survey of Modern Authorship Attribution Methods," *Journal of the American Society for Information Science and Technology,* vol. 60, no. 3, p. 538–556, 2009.

[4] E. Stamatatos, W. Daelemans, B. Verhoeven, M. Potthast, B. Stein, P. Juola, M. A. Sanchez-Perez and A. Barrón-Cedeño, "Overview of the Author Identification Task at PAN 2014," *CLEF 2013 Evaluation Labs and Workshop -- Working Notes Papers,* pp. 877-897, September 2014.

[5] L. Yao and D. Liu, "Wallace: Author Detection via Recurrent Neural Networks," *CS224N Projects,* pp. 1-7, 2015.

[6] I. S. I. Abuhaiba and M. F. Eltibi, "Author Attribution of Arabic Texts Using Extended Probabilistic Context Free Grammar Language Model," *International Journal of Intelligent Systems and Applications,* vol. 8, no. 6, pp. 27-39, 2016.

[7] J. Kapociute-Dzeikiene, L. Sarkute and A. Utka, "The Effect of Author Set Size in Authorship Attribution for Lithuanian," in *Proceedings of the 20th Nordic Conference of Computational Linguistics*, NODALIDA, 2015.

[8] S. Stanko, D. Lu and I. Hsu, "Whose Book is it anyway? Using Machine Learning to Identify the Author of Unknown Texts," 2013.

[9] P. Juola and G. K. Mikros, "Cross-Linguistic Stylometric Features: A Preliminary Investigation," in *International Days of Statistical Analysis of Textual Data*, Nice, 7-10 june, 2016.

[10] U. Sapkota, T. Solorio, M. Montes-y-G´omez, S. Bethard and P. Rosso, "Cross-Topic Authorship Attribution: Will Out-Of-Topic Data Help?," in *the 25th International Conference on Computational Linguistics: Technical Papers*, Dublin, Ireland, August 23-29 2014.

[11] I. Kourtis and E. Stamatatos, "Author Identification Using Semi-supervised Learning Notebook for PAN at CLEF 2011," in *Notebook Papers of CLEF 2011 Labs and Workshops*, Amsterdam, Netherlands,, September 2011.

[12] M. Koppel, J. Schler and S. Argamon, "Computational Methods in Authorship Attribution," *Journal of the American Society for Information Science and Technology,* vol. 60, no. 1, p. 9–26, 2009.

[13] F. Howedi and M. Mohd, "Text Classification for Authorship Attribution Using Naive Bayes Classifier with Limited Training Data," *Computer Engineering and Intelligent Systems,* vol. 5, no. 4, 2014.

[14] S. Swain, G. Mishra and C. Sindhu, "Recent Approaches on Authorship Attribution Techniques-An Overview," in *International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Kancheepuram, Tamil Nadu, India, 2017.

[15] D. Racine, "English Writing Structure, Compared to Other Languages," Magoosh, 27 11 2015. [Online]. Available: https://magoosh.com/toefl/2015/english-writing-structure-compared-to-other-languages/. [Accessed 26 09 202].

[16] B. Gamback, M. Sahlgren, A. Alemu and A. L. Asker, "Applying Machine Learning to Amharic Text Classification," in *5th World Congress of African Linguistics*, Addis Ababa, 7-11 August, 2011.

[17] C. Qian, Tianchang and H. R. Zhang, "Deep Learning based Authorship Identification," Stanford, 2016.

[18] N. INDURKHYA and . F. J. DAMERAU, HANDBOOK OF NATURAL LANGUAGE PROCESSING SECOND EDITION, Boca Raton: Chapman & Hall/CRC, 2010.

[19] P. Juola, "Authorship Attribution," *Foundations and Trends in Information Retrieval,* vol. 1, no. 3, p. 233–334, 2008.

[20] MasterClass, "What Is Prose? Learn About the Differences Between Prose and Poetry With Examples," Master Class, 10 2019. [Online]. Available: https://www.masterclass.com/articles/what-is-prose-learn-about-the-differences-between-prose-and-poetry-with-examples#what-is-the-function-of-prose-in-writing. [Accessed 27 09 2020].

[21] Literary Terms, "Poetry," Literary Terms, 05 11 2015. [Online]. Available: <https://literaryterms.net/>. [Accessed 27 09 2020].

[22] LibApps, "Article Types: What's the Difference Between Newspapers, Magazines, and Journals?," Thomas G. Carpenter Library, [Online]. Available: https://libguides.unf.edu/articletypes. [Accessed 03 11 2020].

[23] L. Asker, A. A. Argaw and B. Gamback, "Classifying Amharic Webnews," *Information Retrieval,* vol. 12, no. 3, pp. 416 - 435, 2009.

[24] S. Eyassu and B. Gambäck, "Classifying Amharic news text using self-organizing maps," in *Semitic '05 Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Ann Arbor, Michigan, 2005.

[25] M. Abate and Y. Assabie, "The Development of Amharic Morphological Analyzer Using Memory Based Learning," in *Ethiopia Information Communication Technology Annual Conference*, Addis Ababa, 2014.

[26] W. Kelemework, "Automatic Amharic text news classification: Aneural networks approach," *Ethiop. J. Sci. & Technol.,* vol. 6, no. 2, pp. 127-137, 2013.

[27] M. L. Bender, Language in Ethiopia, London: Oxford University Press, 1976.

[28] Y. Mohammed, Amharic Grammar and Literature, Addis Ababa: International Leadership Printing Press, 2017.

[29] T. H. GEBERMARIAM, AMHARIC TEXT RETRIEVAL: AN EXPERIMENT USING LATENT SEMANTIC INDEXING (LSI) WITH SINGULAR VALUE DECOMPOSITION (SVD), D. o. C. Science, Ed., Addis Ababa University: Unpublished Masters Thesis, 2003.

[30] Z. SINTAYEHU, Automatic Classification of Amharic News Items, D. o. C. Science, Ed., Addis Ababa University: Unpublished Masters Thesis, 2001.

[31] A. Tefera and Y. Assabie, "Automatic construction of Amharic semantic networks from unstructured text using Amharic wordNet," in *GWC 2014: Proceedings of the 7th Global Wordnet Conference*, 2014.

[32] G. Berhane, "Word formation in Amharic," *Ethiopian Journal of Languages and Literature,* vol. 1, no. 2, pp. 50-74, 1992.

[33] A. A. Argaw and L. Asker, "An Amharic stemmer: Reducing words to their citation forms," in *Proceedings of the 2007 workshop on computational approaches to semitic languages: Common issues and resources*, 2007.

[34] Eriksson, L. Asker and A. Alemu, "An empirical approach to building an Amharic treebank," in *Proc. 2nd Workshop on Treebanks and Linguistic Theories*, 2003.

[35] J. Cowell and F. Hussain, "Amharic character recognition using a fast signature based algorithm," in *Proceedings on Seventh International Conference on Information Visualization*, 2003.

[36] L. Asker, A. A. Argaw, B. Gambäck and M. Sahlgren, "Applying machine learning to Amharic text classification," in *Proceedings of the 5th World Congress of African Linguistics*, 2007.

[37] N. A. Lakew and P. Willett, "The effectiveness of stemming for information retrieval in Amharic," *Program: electronic library and information systems,* vol. 37, no. 4, pp. 254-259, 2003.

[38] A. Gutman and B. Avanzati, "An insatiable appetite for ancient and modern tongues," 2013. [Online]. Available: http://www.languagesgulper.com/eng/Amharic.html. [Accessed 04 11 2020].

[39] F. P. COTTERELL, "Amharic word classes," *Journal of Ethiopian Studies,* vol. 2, no. 1, pp. 33-48, 1964.

[40] G. A. Demeke and M. Getachew, "Manual Annotation of Amharic News Items with Part-of-Speech Tags and its Challenges," 2006.

[41] M. Tachbelie and W. Menzel, "Amharic Part-of-Speech Tagger for Factored Language Modeling," no. 428-433, 2009.

[42] V. N.Gudivada, Handbook of Statistics, Elsevier, 2018.

[43] G. Assefa, Ontology-based Semantic Indexing for Amharic Text in Football, Addis Ababa University: Unpublished Thesis Work, 2013.

[44] E. D. Yirdaw and D. Ejigu, "Topic-based Amharic text summarization with probabilistic latent semantic analysis," in *MEDES '12: Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, 2012.

[45] አ. አለቆሙ, "የአማርኛ ግጥም ባህርዩ፡ መደቡኖ ውበቱ," *Journal of Ethiopian Studies,* vol. 36, no. 2, pp. 5-36, 2020.

[46] ብ. ገበየሁ, " ምጣኔ በአማርኛ ሰነግጥም ተሾመ ይመር አበባሙ," *Journal of Ethiopian Studies,* vol. 33, no. 1, pp. 89-118, 2020.

[47] ተ. ይ. አበባው, "የአማርኛ ግጥም አይነቶች," *Journal of Ethiopian Studies,* vol. 30, no. 1, pp. 89-123, 2020.

[48] S. M. Kotsiantis and T. V. Ikonomakis, "Text Classification Using Machine Learning Techniques," *WSEAS TRANSACTIONS on COMPUTERS,* vol. 4, no. 8, pp. 966-974, 2005.

[49] S. Fabrizio, "Text categorization," in *Encyclopedia of Database Technologies and Applications*, 2005.

[50] D. Harris, D. Wu and V. N. Vapnik, "Support Vector Machines for Spam Categorization," *IEEE TRANSACTIONS ON NEURAL NETWORKS,* vol. 10, no. 5, 1999.

[51] J. Diederich, J. Kindermann, E. Leopold and G. Paass, "Authorship attribution with support vector machines," *Applied intelligence,* vol. 19, no. 1, pp. 109-123, 2003.

[52] S. Argamon, A. R. Shimoni and M. Koppel , "Automatically Categorizing Written Texts by Author Gender," *Literary and Linguistic Computing,* vol. 17, no. 4, 2002.

[53] Y.-B. Lee and S. H. Myaeng , "Text genre classification with genre-revealing and subject-revealing features," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002.

[54] B. Pang, L. Lee and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," in *In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, 2002.

[55] F. SEBASTIANI, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys,* vol. 34, no. 1, pp. 1-47, 2002.

[56] K. Das and R. N. Behera , "A Survey on Machine Learning: Concept,," *International Journal of Innovative Research in Computer,* vol. 5, no. 2, 2017.

[57] J. Qiu, Q. Wu, G. Ding, Y. Xu and S. Feng, "A survey of machine learning for big data," *EURASIP Journal on Advances,* vol. 67, no. 1, 2016.

[58] M. Johannes, M. Marcus and H. Marvin, "A SURVEY OF THE APPLICATION OF MACHINE LEARNING IN DECISION SUPPORT SYSTEMS," in *Twenty-Third European Conference on Information Systems (ECIS)*, Münster, Germany, 2015.

[59] S. Gebeyehu and . D. S. Rao, "A Two Step Data Mining Approach for Amharic Text Classification," *American Journal of Engineering Research (AJER),* vol. 03, no. 04, pp. 251-259, 2014.

[60] M. Koppel, J. Schler and S. Argamon, "Computational methods in authorship attribution," *J. Am. Soc. Inf. Sci. Technol.,* vol. 60, no. 1, pp. 9-26, 2009.

[61] . S. E. M. El Bouanani and I. Kassou , "Authorship Analysis Studies: A Survey," *International Journal of Computer Applications,* vol. 86, no. 12, 2014.

[62] M. S. Tamboli and R. S. Prasad, "Authorship Analysis and Identification Techniques: A Review," *International Journal of Computer Applications,* vol. 77, no. 16, 2013.

[63] T. C. Mendenhall, "The Characteristic Curves of Composition," *American Association for the Advancement of Science,* vol. 9, no. 214, pp. 237-249, 2017.

[64] . M. L. Jockers and D. M. Witten , "A comparative study of machine learning methods for authorship," *Literary and Linguistic Computing,* vol. 25, no. 2, 2010.

[65] P. J. Kumar, G. S. Reddy and T. R. Reddy, "Document Weighted Approach for Authorship Attribution," *International Journal of Computational Intelligence Research,* vol. 13, no. 7, pp. 1653-1661, 2017.

[66] J. Machicao, E. A. Correa Jr, G. H. B. Miranda, D. R. Amancio and O. M. Bruno, "Authorship attribution based on life-like network automata," *PloS one,* vol. 13, no. 3, 2018.

[67] E. Stamatatos, "Authorship attribution using text distortion," 2017.

[68] S. Argamon, C. Whitelaw, P. Chase, S. R. Hota, N. Garg and S. Levitan, "Stylistic text classification using functional lexical features," *Journal of the American Society for information Scienceand Technology,* vol. 58, no. 6, pp. 802-822, 2007.

[69] S. Argamon and S. Levitan , "Measuring the usefulness of function words for authorship attribution," in *In Proceedings of the 2005 ACH/ALLC Conference*, 2005.

[70] . C. Sanderson and S. Guenter , "Short Text Authorship Attribution via Sequence Kernels,Markov Chains and Author Unmasking: An Investigation," in *Conference on Empirical Methodsin Natural Language Processing*, Sydney, Australia, 2006.

[71] H. J. Escalante, T. Solorio and . M. Montes-y-G´omez, "Local Histograms of Character N-grams for Authorship Attribution," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oregon, 2011.

[72] . J. Liu and Y. Zhang , "An Empirical Comparison Between N-gram and Syntactic Language," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015.

[73] F. D. Laramée, "Introduction to stylometry with python," ProgHist Limited, 21 04 201804. [Online]. Available: https://programminghistorian.org/en/lessons/introduction-to-stylometry-with-python. [Accessed 13 12 2018].

[74] D. Mannion and P. Dixon , "Sentence-length and Authorship Attribution: the Case of Oliver Goldsmith," *Literary and Linguistic Computing,* vol. 19, no. 4, 2004.

[75] R. S. and D. . I. H. Forsyth, "Feature-finding for text classification," *Literary and Linguistic Computing,* vol. 11, no. 4, pp. 163-174, 1996.

[76] F. Howedi and M. Mohd , "Text Classification for Authorship Attribution Using Naive Bayes Classifier with Limited Training Data," *Computer Engineering and Intelligent Systems,* vol. 5, no. 4, 2014.

[77] G. Forman, "An Extensive Empirical Study of Feature Selection Metrics for Text Classification," *Journal of Machine Learning Research,* vol. 3, pp. 1289-1305, 2003.

[78] . H. Wang, . X. Zhang and . J. Zhu, "Discrimination-based feature selection for multinomial naïve bayes text classification," in *In International Conference on Computer Processing of Oriental Languages*, Berlin, Heidelberg, 2006.

[79] H. Zou, T. Hastie and R. Tibshirani, "Sparse principal component analysis," *Journal of computational and graphical statistics,* vol. 15, no. 2, pp. 265-286, 2006.

[80] . C. E. Chaski, "Who's At The Keyboard? Authorship Attribution in Digital Evidence Investigations," *International Journal of Digital Evidence,* vol. 4, no. 1, 2005.

[81] O. Uzuner, B. Katz and T. Nahnsen, "Using Syntactic Information to Identify Plagiarism," in *Proceedings of the 2nd Workshop on Building Educational Applications Using NLP*, Ann Arbor, 2005.

[82] L. Wang and V. Kecman , Support vector machines: theory and applications, 2005.

[83] J. J. Rodriguez , L. I. Kuncheva and C. . J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE transactions on pattern analysis and machine intelligence,* vol. 28, no. 10, pp. 1619-1630, 2006.

[84] S. Argamon, "Interpreting Burrows's Delta:Geometric and Probabilistic," *Literary and Linguistic Computing,* vol. 23, no. 2, pp. 131-147, 2008.

[85] E. Keogh, C. A. Ratanamahatana, S. Lonardi, . L. Wei, S.-H. Lee and J. Handley, "Compression-based data mining of sequential data," *Data Mining and Knowledge Discovery,* vol. 14, no. 1, pp. 99-129, 2007.

[86] M. Koppel, J. Schler and E. Bonchek-Dokow, "Measuring differentiability: Unmasking pseudonymous authors," *Journal of Machine Learning Research,* pp. 1261-1276, 2007.

[87] H. V. Halteren, "Author verification by linguistic profiling: An exploration of the parameter space," *ACM Transactions on Speech and Language Processing (TSLP),* vol. 4, no. 1, p. 1, 2007.

[88] H. Gómez-Adorno, G. Sidorov, D. Pinto and . I. Markov, "A graph based authorship identification approach," in *Working notes papers of the CLEF*, 2015.

[89] V. Q. Marinho, G. Hirst and D. R. Amancio, "Authorship attribution via network motifs identification," in *In 2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*, 2006.

[90] V. Q. Marinho, G. Hirst and D. R. Amancio, "Labelled network motifs reveal stylistic subtleties in written texts," in *arXiv preprint arXiv:1705.00545*, 2017.

[91] D. R. Amancio, "A complex network approach to stylometry," *PloS one,* vol. 10, no. 8, 2015.

[92] S. Lahiri and R. Mihalcea, "Authorship attribution using word network features," in *arXiv preprint arXiv:1311.2978*, 2013.

[93] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel and M. Blondel, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

[94] R. Jain, "introduction-naïve bayes algorithm codes python-r," HackerEarth, 02 02 2017. [Online]. Available: https://www.hackerearth.com/blog/developers/introduction-naive-bayes-algorithm-codes-python-r/. [Accessed 25 03 2019].

[95] R. Nisbet, K. Yale and G. Miner, Handbook of Statistical Analysis and Data Mining Applications, Elesvier, 2018.

[96] L. Tanguy, A. Urieli, B. Calderone, N. Hathout and F. Sajous, "A Multitude of Linguistically-rich Features for Authorship Attribution," in *PAN Lab at CLEF*, Amsterdam, Netherlands, 2011.

[97] G. P. Zhang, "Neural Networks for Classification: A Survey," *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS,* vol. 30, no. 4, 2000.

[98] A. Khatun, A. Rahman, M. S. Islam and M.-E.-J. E-Jannat, "Authorship Attribution in Bangla literature using Character-level CNN," in *22nd International Conference on Computer and Information Technology (ICCIT)*, 2019.

[99] M. Sokolova and G. Lapalme , "A systematic analysis of performance measures for classification tasks," *Information Processing and Management,* vol. 45, p. 427–437, 2009.

[100] S. Raghavan, A. Kovashka and R. Mooney, "Authorship attribution using probabilistic context-free grammars," in *In Proceedings of the ACL 2010 conference short papers*, 2010.

[101] S. Mechti, M. Jaoua, R. Faiz and L. H. Belguith, "On the Empirical Evaluation of Author Identification Hybrid Method Notebook for PAN at CLEF 2015," in *CLEF 2015 Evaluation Labs and Workshop – Working Notes Papers*, Toulouse, France, 2015.

[102] A.-F. Ahmed, R. Mohamed and B. Mostafa, "Machine Learning for Authorship Attribution in Arabic," *International Journal of Future Computer and Communication,* vol. 6, no. 2, 2017.

[103] S. PK, "Applicability of Relevant Authorship Attribution Technique in Malayalam," *International Journal of Innovative Research in Science, Engineering and Technology,* vol. 6, no. 2, 2017.

[104] R. S. Silva, L. Sarmento, . T. Grant, E. Oliveira and B. Maia, "Comparing sentence-level features for authorship analysis in portuguese," in *In International Conference on Computational Processing of the Portuguese Language*, Berlin, Heidelberg, 2010.

[105] N. Kaur and A. Verma, "Authorship Attribution of Punjabi Poetry using SVM Classifier," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 5, no. 5, May 2015.

[106] F. S. a. R. D. Hamid Tizhoosh, "Poetic Features for Poem Recognition: A Comparative Study," *JOURNAL OF PATTERN RECOGNITION RESEARCH,* vol. 3, pp. 24-39, 2008.

[107] T. Andargie, INCORPORATING LINGUISTIC FEATURES IN BI-DIRECTIONAL AMHARIC - ENGLISH STATISTICAL MACHINE TRANSLATION, a. a. u. School of Information science, Ed., Addis Ababa: Unpublished Masters Thesis, 2018.

[108] M. Gasser, "HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya," 2011.

[109] L. Marie, Sweet 'N Spicy Designs, 31 05 2015. [Online]. Available: http://lzmarieauthor.com/syntax-smarts/. [Accessed 21 01 2019].

[110] Y. F. a. D. Yaqob, "The System for Ethiopic Representation in ASCII," 1997.

[111] MasterClass, "What Is Prose? Learn About the Differences Between Prose and Poetry With Examples," Master Class, 22 10 2019. [Online]. Available: https://www.masterclass.com/articles/what-is-prose-learn-about-the-differences-between-

prose-and-poetry-with-examples#what-is-the-function-of-prose-in-writing. [Accessed 27 09 2020].

[112] Literary Terms, "Poetry," Literary Terms, 05 11 2015. [Online]. Available: <https://literaryterms.net/>. [Accessed 27 09 2020].

Annexes

Annex A: Class (Author) Names

No	Class (Author) Name
1	Berhanu Fekade
2	Biruk Abdu
3	Dawit Endeshaw
4	Dawit Taye
5	Dawit Tolossa
6	Dereje Tegenaw
7	Fissaha Getnet
8	Henok Yared
9	Kaleyesus Bekele
10	Mihret Moges
11	Nathan Dawit
12	Neamn Ashenafi
13	Samson Abdela
14	Shahida Hussen
15	Tadesse Gebremariam
16	Tamiru Tsige
17	Tamrat Getachew
18	Widneh Zenebe
19	Yohannes Enberbr
20	Zemenu Tenagne

No	Class (Poet) Name
1	Bewketu Seyoum
2	Gebrekristos Desta

Annex B: The Amharic Alphabet

1st order	2nd order	3rd order	4th order	5th order	6th order	7th order
ሀ hä	ሁ hu	ሂ hi	ሃ ha	ሄ he	ህ h	ሆ ho
ለ lä	ሉ lu	ሊ li	ላ la	ሌ le	ል l	ሎ lo
ሐ hä	ሑ hu	ሒ hi	ሓ ha	ሔ he	ሕ h	ሖ ho
መ mä	ሙ mu	ሚ mi	ማ ma	ሜ me	ም m	ሞ mo
ሠ sä	ሡ su	ሢ si	ሣ sa	ሤ se	ሥ s	ሦ so
ረ rä	ሩ ru	ሪ ri	ራ ra	ሬ re	ር r	ሮ ro
ሰ sä	ሱ su	ሲ si	ሳ sa	ሴ se	ስ s	ሶ so
ሸ šä	ሹ šu	ሺ ši	ሻ ša	ሼ še	ሽ š	ሾ šo
ቀ qä	ቁ qu	ቂ qi	ቃ qa	ቄ qe	ቅ q	ቆ qo
በ bä	ቡ bu	ቢ bi	ባ ba	ቤ be	ብ b	ቦ bo
ተ tä	ቱ tu	ቲ ti	ታ ta	ቴ te	ት t	ቶ to
ቸ čä	ቹču	ቺ či	ቻ ča	ቼ če	ች č	ቾ čo
ኀ hä	ኁ hu	ኂ hi	ኃ ha	ኄ he	ኅ h	ኆ ho
ነ nä	ኑ nu	ኒ ni	ና na	ኔ ne	ን n	ኖ no
ኘ ňä	ኙ ňu	ኚ ňa	ኛ ňa	ኜ ňe	ኝ ň	ኞ ňo
አ ä	ኡ u	ኢ i	ኣ a	ኤ e	እ □	ኦ o
ወ wä	ዉ wu	ዊ wi	ዋ wa	ዌ we	ው w	ዎ wo
ዐ ä	ዑ u	ዒ i	ዓ a	ዔ e	ዕ □	ዖ o
ከ kä	ኩ ku	ኪ ki	ካ ka	ኬ ke	ክ k	ኮ ko
ኸ hä	ኹ hu	ኺ hi	ኻ ha	ኼ he	ኽ h	ኾ ho
ዘ zä	ዙ zu	ዚ zi	ዛ za	ዜ ze	ዝ z	ዟ zo
ዠ žä	ዡ žu	ዢ ži	ዣ ža	ዤ že	ዥ ž	ዧ žo

ይ yä	ዩ yu	ዪ yi	ያ ya	ዬ ye	ይ y	ዮ yo
ግ gä	ጉ gu	ጊ gi	ጋ ga	ጌ ge	ግ g	ጎ go
ድ dä	ዱ du	ዲ di	ዳ da	ዴ de	ድ d	ዶ do
ጅ jä	ጁ ju	ጂ ji	ጃ ja	ጄ je	ጅ j	ጆ jo
ጠ ṭä	ጡ ṭu	ጢ ṭi	ጣ ṭa	ጤ ṭe	ጥ ṭ	ጦ ṭo
ፌ ćä	ፉ ću	ፊ ći	ፋ ća	ፌ će	ፍ ć	ፎ ćo
ሸ şä	ሹ şu	ሺ şi	ሻ şa	ሼ şe	ሽ ş	ሾ şo
ፀ ṣä	ፁ ṣu	ፂ ṣi	ፃ ṣa	ፄ ṣe	ፅ ṣ	ፆ ṣo
ጰ p̈	ጱ p	ጲ ṗ	ጳ p̈	ጴ p	ጵ ρ	ጶ ρ
ፍ fä	ፉ fu	ፊ fi	ፋ fa	ፌ fe	ፍ f	ፎ fo
ፕ pä	ፑ pu	ፒ pi	ፓ pa	ፔ pe	ፕ p	ፖ po

Annex C: Amharic Punctuation Marks

Punctuation	Amharic Name
.	አንድ ነጥብ (aned netib)
:	ሁለት ነጥብ (hulet netib)
:-	ሁለት ነጥብ ከሰረዝ (hulet netib keserez)
…	ሦስት ነጥብ (sost netib)
፣	ነጠላ ሰረዝ (netela serez)
፤	ድርብ ሰረዝ (dereb serez)
«»	ትዕምርተ ጥቅስ (teemirte teqs)
!	ትዕምርተ አንክሮ (teemirte ankro)
¡	ትዕምርተ ስላቅ (teemirte selaq)
?	የጥያቄ ምልክት (yetyaqe milkt)
/	እዝባር (ezbar)
()	ቅንፍ (qenef)
-	ንዑስ ሰረዝ (neus serez)
__	ዐብይ ሰረዝ (abiy serez)
=	ዕሩይ (eruy)
*	ረድፍ (redf)
^	እመጫት (emechat)
::	አራት ነጥብ (arat netib)

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of materials used for the thesis have been dully acknowledged.

Declared by:

Name: _____

Signature: _____

Date: _____

Confirmed Advisor:

Name: _____

Signature: _____

Date: _____

Addis Ababa, November 2020