University of Leipzig
Faculty of Mathematics and Computer Science
BSc. Computer Science

On the Viability of Phonetic Transcriptions for Authorship Verification

Bachelor's Thesis

David Reinartz

Referee: Prof. Dr. Martin Potthast

Submission date: August 8, 2021

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Leipzig, August 8, 2021

.....
David Reinartz

Abstract

Authorship Verification is the task of deciding whether two texts were written by the same author or by different authors. We hypothesize that authors have a *phonetic preference*, based on which they produce texts, and that we can use this phonetic information to aid in classification. Using a range of phonetic transcription systems of different granularity, we examine the viability of using transcription-based features in two well-known Authorship Verification algorithms. We find that the use of phonetic representations of text does not yield an improvement in performance. In fact, for many configurations we record statistically significant decreases in performance. We propose three possible explanations for the negative results. For reproducibility, all code is published as open-source.

Contents

Chapter 1

Introduction

With the widespread availability and use of text as a medium of information transfer, the problem of identifying authorship of given texts has become one of the main focuses of stylometric analysis. In this report, we specifically tackle the task of Authorship Verification which consists of classifying whether two given texts were written by the same author or not. Underlying our approach is the hypothesis that authors have a *phonetic preference*, based on which they produce different texts. Ladefoged and Johnson [2014] titles this preference the "phonetics of the individual" and states that "[...] the set of phonetic habits and memories that each speaker possesses is different from those of every other speaker of the language". By applying phonetic transcription systems of varied granularity to the data used, we aim to emphasize these phonetically relevant features implanted into the texts by their authors. Then, we use the transcribed data to train two well-known Authorship Verification classifiers. By evaluating the results with standard measures used in Authorship Verification, we aim to answer the following questions:

- Does the prior phonetic transcription of texts improve the performance of the algorithms over using verbatim text?
- Are the results correlated to the granularities of the transcriptions?

We transcribe textual data to North American English pronunciation. We are aware that in this process some author-specific information is lost and will discuss the implications of this. Nevertheless, we anticipate that the phonetic information encoded into plain text and boosted by transcribing gives the classifiers an advantage over more naïve methods. All code attributions are open-sourced at `https://github.com/av-pt/bachelors-thesis` with an emphasis on ease of reproduction.

Chapter 2

Background

2.1 Authorship Verification

In our world a large amount of communication and information transfer is done through a visual medium: text. This was not always the case. According to Roser and Ortiz-Ospina [2016], for most of history, reading and writing was reserved to an elite and associated with power. Through the spread of public education over time, increasingly more people were able to read and write. World-wide literacy rates are estimated to have been at 12% in 1820 and have risen to 86% in 2016. Buringh and Van Zanden [2009] estimates that the number of books produced annually in Western Europe in the sixth and seventh centuries was only around 120. In contrast, and not only due to the invention of letterpress printing with movable type, the production in 1790 was at a peak of 20,000,000 books per year. These historical processes gave rise to the linguistic branch of stylometry, the analysis of "an author's work based especially on the recurrence of particular turns of expression or trends of thought"[1]. The worth of profiling authors can be illustrated with an anecdote about a court case from 1979 published in Hitt [2012]. The American linguist Roger Shuy was able to infer that the author of a ransom note linked to a kidnapping case had to be well-educated and from Akron, Ohio. Using this information, the offender was caught and later confessed the crime.

With the adaption of computers by the linguistic community, stylometry was increasingly done automatically. As presented in Stamatatos [2009], the wide availability of training data and the increasing speed of computers allowed for more involved and complex stylometric methods. Nowadays, Authorship Analysis, concerned mainly with determining authors of given texts, is consolidated as a subbranch of stylometry. According to Bevendorff et al. [2020b], it

[1] `https://www.merriam-webster.com/dictionary/stylometry`

can be further parted into four disciplines:

Authorship Attribution aims at selecting the author of a given text document from a list of candidate authors. This task simulates, for example, a forensic situation where the document is a threatening letter, and it has to be determined which of the possible suspects created it. *Authorship Obfuscation* changes the direction of the effort. Instead of trying to determine an author, the goal is to paraphrase a given text in such a way that its author cannot be identified by comparing it to other non-paraphrased texts of the same author. For example, the author of said threatening letter could actively try to use different language to conceal their identity. Although it seems at first counter-intuitive to model this behavior, by developing working obfuscation methods mistakes and pitfalls in the creation of Authorship Analysis methods can be uncovered. To this end, *Obfuscation Evaluation* measures are devised which assess the performance of the obfuscation methods. Lastly, *Authorship Verification* (AV), which is the task of interest to our research, is defined in Bevendorff et al. [2019a] as: "[G]iven a pair of documents, determine whether they are written by the same author". This task was first proposed in 2004 by Koppel and Schler, and thus is a rather recent development. Authorship Verification abstracts on other Authorship Analysis tasks by employing only basic inputs and outputs. In contrast to Authorship Attribution, it does not aim to identify actual authors but only the fact of whether they are different or not. This arguably makes Authorship Verification a more difficult task, since no additional information about the authors of the two texts can be used to extract author profiles from. Instead, Authorship Verification algorithms try to identify the universal characteristics by which authors can be differentiated, regardless of the style of one specific author. As stated in Bevendorff et al. [2020b], the PAN[2,3] series of shared tasks included Authorship Verification tasks from 2013 to 2015 and picked them back up with PAN 2020, with a planned total of three tasks over the years 2020 to 2022. In the course of reintegrating the task to PAN 2020, a new data format has been devised, which we will use in our research.

Authorship Verification can be formalized as follows. Given a text pair (d_1, d_2), classify it to $True, False$, i.e., approximate the target function $\phi : (d_1, d_2) \rightarrow \{True, False\}$ where $\phi(d_1, d_2) = True$ iff d_1 and d_2 have the same author. We call an algorithm that approximates ϕ an Authorship Verification classifier, or AV classifier for short. In practice, the output of a binary classifier is often a real number in the interval $[0, 1]$, signifying the probability that a

[2] `https://pan.webis.de/`

[3] Wikipedia: "originally, plagiarism analysis, authorship identification, and near-duplicate detection, later more generally workshop on uncovering plagiarism, authorship, and social software misuse", `https://en.wikipedia.org/wiki/Stylometry#PAN`

sample belongs to the positive or *True* class. As discussed later, the precision of an AV classifier is arguably more important than its recall. Because of this, PAN 2020 uses metrics that take *non-answers* into account. An AV classifier can decide that the chance of a same-author classification is not high enough and withhold an answer. Usually, this is implemented with an uncertainty interval around 0.5 for which *non-answer* is returned.

2.2 Phonetic Transcriptions

As defined in O'Grady et al. [2017], which serves as a reference for the notions in this section, phonetics[4] is the branch of linguistics concerned with "the inventory and structure of the sounds of speech". Not all sounds humans can articulate are present in the worlds languages. Yet a wide range, estimated to consist of 600 consonant and 200 vowel sounds, occur in human language. Note that phonetics is different from phonology, in that phonology examines how sounds create meaning in a language. Two subbranches of phonetics are articulatory phonetics and acoustic phonetics. The former concerns itself with the physiological processes involved in speech production while the latter examines acoustic characteristics of speech. In this report we focus on articulatory phonetics.

The distinct sounds of which a spoken utterance is made up are called *phones*. On a more abstract level, linguists segment speech into *phonemes*. Phonemes are defined as the smallest units of sound distinguishing meaning in the words of a language. Swapping a phoneme in a word for another one changes its meaning, while replacing a phone with a different one does not necessarily alter its meaning. Those sets of phones that do not evoke a change of meaning when exchanged are called *allophones* of their respective phonemes. For example, the alveolar nasal consonant [n][5] and the dental nasal consonant [n̪] are allophones of the phoneme /n/[6] as they are not used to differentiate meaning in English — [wʌn] and [wʌn̪] both point to the same concept "one". However, the alveolar nasal consonant [n] and the bilabial nasal consonant [m] are different, contrasting phonemes — [mæp] and [næp] indicate the two distinct concepts "map" and "nap". While phones are universal, phonemes are language specific.

As hypothesized in the introduction, we suspect that information valuable for identifying authorship exists on the phonetic level. Because Authorship

[4]Merriam-Webster: from Greek phōnētikós, "of speech, endowed with speech" , `https://www.merriam-webster.com/dictionary/phonetics`

[5]Square brackets are used to generally mark phonetic notation in IPA.

[6]Slashes are used to mark an abstract phonemic notation in IPA, omitting details that would be used to distinguish sounds in the specific language notated.

Table 2.1: Example transcriptions.

System	Transcription
Verbatim	Wake and rise, and step into the green outdoors.
IPA	weɪk ʌnd ɹaɪz ʌnd stɛp ɪntu ðʌ gɹin aʊtdɔɹz
Dolgo	WVK VNT RVS VNT STVP VNTV TV KRVN VTTVRS
ASJP	wek ond raz ond stEp intu 8o grin atdorz
CV	CVC VCC CVC VCC CCVC VCCV CV CCVC VCCVCC
Soundex	W200 A530 R200 A530 S310 I530 T000 G650 O362
RefSoundex	W030 A086 R9030 A086 S3601 I0860 T60 G4908 O06093
Metaphone	WK ANT RS ANT STP INT 0 KRN OTTRS

Verification classifiers use text as input, we want to utilize methods to extract these phonetically relevant features from the texts. One possible way of achieving this is by employing phonetic transcriptions. For our purposes, these are transformations assigning a symbol to each sound of a text as if the text was spoken aloud. Phonetic transcriptions can be seen as data reduction methods. By applying them, we anticipate that the phonetically relevant features stay apparent while other, less relevant features stand out less. In total, we use eight phonetic transcription systems of different granularity. The *narrower* a transcription, the more closely it follows the phonetic details of an utterance. This often leads to the system having a bigger alphabet, such as the IPA described below. The *broader* a transcription, the more it generalizes phonetic features. Table 2.1 shows example transcriptions for one of the "phonetically balanced" sentences developed by the IEEE [1969].

The most widely used phonetic transcription system is the International Phonetic Alphabet (**IPA**). As outlined in the Handbook of the International Phonetic Association [1999], it was developed by the International Phonetic Association founded in 1886. It serves as a system to notate the sounds of languages in an internationally agreed-upon manner. Pulmonic consonants — consonants initiated by a buildup of pressure from the lungs — are distinguished in their place and manner of articulation. The place of articulation describes the position in the vocal tract where the sound is produced. For example, a bilabial sound, such as the "b" in "beer", is articulated with both lips whereas a glottal sound, such as the "h" in "hello", is articulated all the way back at the glottis. The manner of articulation includes several factors regarding distinctive ways of sound production. To give an example, a plosive, such as the "p" in "explosion", is created by completely stopping the airflow, building up pressure and suddenly releasing said pressure. The IPA also differentiates between voiced and voiceless consonants such as the first phonemes

in the words "vast" and "fast". In a similar way, non-pulmonic consonants and vowels are organized on scales such as position and manner of articulation. This way, a system to classify arbitrary sounds of a language has been created. With 155 symbols, its alphabet is the largest of the transcription systems considered in this report. Therefore, the produced transcriptions are usually the narrowest. It should be noted that when using the IPA system a transcription can be much more detailed than just using the correct symbols for the phonemes. Using diacritics, many other qualities of speech can be indicated, for example, to transcribe specific dialects. Creating accurate and detailed transcriptions of a given speech sample on the level of phones is a difficult task usually done manually by linguists. This ties into a problem we have found in our research that we will discuss later on. To achieve more stable results, we use a slightly broader version of the IPA omitting prosodic markers and diacritics.

Because of their detailed nature, IPA transcriptions contain a lot of information. Continuing with the idea of reducing phonetically irrelevant information, we also employ broader transcription systems. The following ones can be categorized as sound class systems organizing speech sounds into linguistically-informed classes.

According to List [2012], the term "sound class" was first devised and detailed in Dolgopolsky [1986]. For conciseness, we will use the term more generically as defined above. In the Dolgopolsky sound class system (**Dolgo**), phonemes are organized into ten classes, so that the difference between sounds inside a class is smaller than the difference between classes. The classes were derived manually from empirical data. We use a slightly extended version of the original *Dolgo* sound class system, as implemented in Anderson et al. [2018]. It includes an eleventh class for vowels and is compatible with all IPA symbols including common diacritics. A list of the *Dolgo* sound classes with examples for corresponding phonemes can be seen in Table 2.2.

The Automated Similarity Judgment Program is a project aiming to classify the world's languages introduced in Brown et al. [2008]. As of August 8, 2021, it consists of a database comprising 40-word lists of core vocabulary translated to 9,788 languages. The word lists include meanings such as "I", "drink", and "water". Each word is transcribed using the asjpCode transcription system. This way, phonetic similarities between language pairs can be computed. Language-similarity-trees created with ASJP produce near expert-level classifications. AsjpCode (**ASJP**) consists of 34 consonant and 7 vowel symbols. It can be seen as a simplified variant of the IPA system, with some symbols representing a broader class of speech sounds. For example, "N" represents the velar nasal [ŋ] directly, while "o" represents all rounded

Table 2.2: *Dolgo* sound classes, adopted from List [2010] with the eleventh category "V" added.

Symbol	Example phonemes	Category
P	p, b, f	labial obstruents
T	d, t, θ, ð	dental obstruents
S	s, z, ʃ, ʒ	sibilants
K	k, g, ts, tʃ	velar obstruents, dental and alveolar affricates
M	m	labial nasal
N	n, ɲ, ŋ	remaining nasals
R	r, l	liquids
W	v, u	voiced labial fricative and initial rounded vowels
J	j	palatal approximant
H	h, ɦ	laryngeals and initial velar nasal
V	ɑ, ɛ, ɪ	other vowels (simple and diphtongs)

and unrounded mid and low back vowels [ɤ, ʌ, ɑ, o, ɔ, ɒ]. Another benefit of asjpCode, although not directly influential to our research, is that it consists of only those symbols which are found on a standard QWERTY keyboard. This facilitates manual transcription.

Lastly, the **CV** sound class system assigns the symbol "C" to consonant phonemes and the symbol "V" to vowel phonemes as done in List et al. [2017]. With a binary alphabet it is the broadest of the transcription systems we use.

Apart from these systems we also examine the impact of three simple phonetic algorithms. These algorithms were invented to match words of similar pronunciation in English. The **Soundex** algorithm, patented by Robert C. Russell in 1918 and 1922, was devised for indexing names. By grouping names by phonetic similarity instead of alphabetically, the time needed to search for a given name would be shortened. Also, similar-sounding names that are written differently would be organized into the same categories simplifying access when, for example, only a spoken name is given. A word is represented by a code consisting of a capital letter, the first character of the word, and three digits. The digits, ranging from 1 to 6, represent sound classes of letters occurring later in the word. Table 2.3 shows these classes in more detail. The process of assigning these codes roughly functions as follows. The first letter in the word is used as the beginning letter of the code. The first letter along with all occurrences of the letters "a", "e", "i", "o", "u", "y", "h", and "w" are removed. The remaining letters are encoded using the mapping from Table 2.3. If two equal sound classes appear next to each other, the second occurrence

Table 2.3: Soundex sound classes as used in our research.

Symbol	Associated characters	Category
1	b, f, p, v	labials, labio-dentals
2	c, g, j, k, q, s, x, z	gutturals, sibilants
3	d, t	dental-mutes
4	l	palatal-fricatives
5	m, n	nasals
6	r	dental-fricatives

Table 2.4: Refined Soundex sound classes as used in our research.

Symbol	Associated characters
0	a, e, i, o, u, y, h, w
1	b, p
2	f, v
3	c, k, s
4	g, j
5	q, x, z
6	d, t
7	l
8	m, n
9	r

is removed. The resulting code is truncated to a length of four characters in total. If the code is shorter than four characters it is filled up with trailing zeros.

The Refined Soundex algorithm (**RefSoundex**) extends its predecessor, with the main difference being that the resulting codes are no longer truncated or extended to a length of 4 but instead retain their original length. Also, the number of the sound classes is increased to nine, instead of six, leading to a narrower transcription. The alternate mapping can be seen in Table 2.4. Lastly, the digit sequence following the first character also includes this character's sound class symbol. The word "and", for example, is transcribed to "A086", not "A86". Howard and James [2019] traces the origins of Refined Soundex back to an implementation in the Apache Commons Library as noted in Fossati and Di Eugenio [2008], but indicates that the idea of modifying the sound classes already appeared in Zobel and Dart [1995].

Metaphone is also a phonetic indexing algorithm first published in Philips [1990]. It improves on the Soundex family of algorithms by taking a larger number of inconsistencies and edge-cases of English pronunciation into ac-

count. Also, its focus does not only lie on indexing names but rather English words in general. It consists of a series of 27 context-aware transformations[7], sequentially replacing phonetically similar patterns with representative symbols or removing them if they are not unpronounced. For example, one such transformation is removing the first letter of a word if that word starts with "KN", "GN", "PN", "AE", or "WR". *Metaphone*'s alphabet consists of 21 symbols — 16 for consonants and 5 for vowels — representing classes of speech sounds. Vowel symbols only appear at the beginning of transcribed words. Metaphone was later superseded by Double Metaphone and the closed-source Metaphone 3, both of which use a substantially larger rule set.

For normalization, we remove all inter-word punctuation in the texts. Intra-word punctuation, such as in "don't", is phonetically significant and thus not removed. For brevity, we refer to all systems described above as phonetic transcription systems. In addition to the phonetic transcriptions above, we also create three other conversions for comparison:

- **P**: Removing punctuation
- **PL**: Removing punctuation and lemmatizing the occurring words
- **PLS**: Removing punctuation, lemmatizing, and removing stop words

We handle *verbatim* text and the three non-phonetic conversions in the same way as transcribed text.

[7]The number of transformations stems from the implementation used.

Chapter 3

Related Work

Before the inception of Authorship Verification as a task, Authorship Attribution was the main subject of stylometric investigation, as it is much closer to real-life circumstances. Stamatatos [2009] divides the scientific efforts on Authorship Attribution into two periods — before and after the late 1990s — and gives an overview of their development.

The first application of statistics to authorship research was done in 1887 and 1901 by Mendenhall. Histograms of word length frequencies were used to differentiate Shakespeare from other authors of his era. Mendenhall's work was later critized by Williams [1975] as the differing histograms could better be explained by differences in presentation — Shakespeare used verse while the other authors wrote prose. The most influential study of this early period was conducted by Mosteller and Wallace in 1964 on "The Federalist Papers"[1]. It employed a Bayesian statistical analysis on a small set of common words showing significant capabilities in distinguishing candidate authors. Whereas before, Authorship Attribution was mainly conducted manually by experts, this study paved the way for statistically supported methods. Nevertheless, research continued to mainly focus on solving specific literary disputes and finding new measures to quantify stylometric features. Stamatatos states that the evaluation of the emerging methods was hindered mainly by datasets being too small, unstandardized, and unhomogenized for style and topic. This prevented meaningful comparisons of the different approaches during this first period.

Through the expanding use of Internet media during the late 1990s, data utilized in Authorship Attribution began shifting towards electronic texts. Following this trend, research efforts began to focus on the development of applications that could be used in real-world scenarios such as forensics or law. Additionally, the evaluation of proposed methods was emphasized to enable

[1] `https://en.wikipedia.org/wiki/The_Federalist_Papers`

an objective comparison between them.

On a technical level, Stamatatos splits the proposed approaches into two components: the features used to quantify writing style, and the methods used to attribute authorship. The stylometric features vary in complexity, ranging from lexical features working on word level up to semantic features aiming at extracting the meaning of a text. The most notable *lexical* feature set is the set of most common words. Although for semantic analyses these so-called function words (articles, prepositions, etc.) do not carry much information, they are well suited for discriminating between authors (Argamon and Levitan [2005], Burrows [1987]). The Unmasking algorithm by Koppel and Schler [2004], described in more detail in Section 5.2, uses the 250 most common words of the supplied data as features. On *character* level, many approaches employ character n-grams, reporting very good results (Peng et al. [2003], Kešelj et al. [2003], Stamatatos et al. [2006]). In contrast to lexical features working with words as atomic and isolated units of information, character-based features such as n-grams can take advantage of subword and context information. Because of this, we also use n-grams in our research. The compression-based approach by Teahan and Harper [2003], described in greater detail in Section 5.1, can also be interpreted as a character level approach, as the internal compression algorithm works with characters as atomic units. Commonly used *syntactic* features include *part-of-speech* (POS) frequencies and n-grams. Using a POS-tagger, syntactic information is annotated, signaling if a word is, for example, a noun or a preposition. The resulting tags or sequences thereof are counted, and their frequencies used as features. On a higher level, *semantic* features are used. Most noteworthy is the approach by Argamon et al. [2007]. It extracts semantic information by mapping certain keywords and phrases in specific part-of-speech contexts to semantic meanings. The word "while", for example, is semantically tagged as a *conjunction* that could be followed by an *elaboration*, an *extension*, or an *enhancement* of the previous statement. The frequencies of these semantic phenomena are then used as features for classification.

In general, the different features function only to the extent that their underlying Natural Language Processing algorithms are robust. Low-level features such as character n-grams are trivial to generate from given text and thus produce reliable output feature quality. High-level features like semantic analyses, on the other hand, require much more effort and depend on factors such as target language and corpus quality.

The methods of attributing authorship can further be divided into profile-based and instance-based approaches. The former concatenates the texts by each author into one file, creating a profile of that author. The latter treats each text as a separate instance from which the attribution model

can be trained. As we examine Authorship Verification in our research, the approaches we use are instance-based. However, it can be noted that the compression-based approach was originally developed for Authorship Attribution and merged the texts of each author into one document, rendering it a profile-based approach. As we do not have information about the authors of the given texts, we use an adaption of the compression-based approach that does not create such profiles and simply compares two given texts.

After 2008, through the PAN workshops in 2013–15 and 2020 onward, numerous approaches aiming at solving Authorship Verification were contributed. The most notable recent development is the inception of approaches using deep learning techniques which was enabled by the introduction of a large dataset in PAN 2020 (Boenninghoff et al. [2020], Weerasinghe and Greenstadt [2020], Araujo-Pino et al. [2020], Ordoñez et al. [2020]).

Bevendorff et al. [2019a] reveals possible biases ($B1$–$B6$) in Authorship Verification and presents ways to mitigate these. First, the paper discusses biases of AV classifiers:

- $B1$: Models using corpus-relative features such as TF-IDF are prone to overfitting as in most cases the document frequencies are derived from the training sets themselves.
- $B2$: In a similar vein, models employing feature scaling also tend to overfit to the specifics of the training set. Thus, care should be taken to avoid modelling the training data too closely.

Next, biases concerning the data are examined:

- $B3$: A text may contain artifacts that were not introduced by the author, such as editorial marks or text conversion errors. To prevent fitting to erroneous artifacts, texts should be fully homogenized to only contain artifacts entered by the author.
- $B4$: To increase the size of a dataset, text chunks are often reused. This should not be done, as the resulting corpus might over- or underrepresent certain authors' styles.
- $B5$: Reusing text might lead to overlap including topic words, named entities and other unique character sequences. To inhibit an AV algorithm learning these features, text overlap should be analyzed and corrected.

Lastly, a bias appearing in the evaluation phase is identified:

- $B6$: it is unrealistic for an AV algorithm to be used in situations where it has access to a large test set. Therefore, while evaluating the algorithms should only have access to one text pair at a time. This more closely models manual Authorship Verification where a forensic linguist also inspects text pairs on a case-by-case basis.

To mitigate the biases stemming from the data, a corpus containing texts from Project Gutenberg is presented. We will use this dataset in our experiments.

Research combining Phonetics and Authorship Analysis is sparse. As known to the author, Khomytska et al. conducted the only research on this topic. Khomytska et al. [2019] analyzes the influence of eight different consonant phoneme classes in differentiating authorial style. The consonant phoneme classes that are used group labial, velar, fricative, nasal, sonorous, coronal, dorsal, and stop phonemes. First, a text pair is transcribed and then processed to yield a sample of 51,000 consonant phonemes for each text. The sample is divided into 51 parts and the mean frequencies of the classes are calculated. Using Pearson's test, it is proven that the obtained class frequencies follow a normal distribution. To assess the similarity of the distributions, the Student's t-test, the Kolmogorov-Smirnov test, and the Chi-square test are examined. Also, by comparing the phoneme class frequencies between the texts, differentiation capabilities for each of the classes are determined. It is concluded that labial, fricative, nasal, coronal, dorsal and stop consonant phonemes in conjunction with the Kolmogorov-Smirnov test are useful for differentiating authorial style, whereas velar and sonorous consonant phonemes are not. Khomytska et al. have published a number of similar articles on Style Differentiation and Authorship Attribution. Unfortunately, no standard evaluation methods are used, preventing meaningful comparisons to other work in this area. In addition, the used datasets are small, with the paper outlined above deriving a not further specified improvement in the differentiation of authorial styles by analyzing only one text pair, questioning the validity of the results.

Phonetic transcriptions have also been used as classification features before, namely in the task of Native Language Identification in Smiley and Kübler [2017]. Given a text, the goal is to determine the native language of the author from a closed set of possible languages. Labeled texts from a training set were transcribed using one of four algorithms. Three of the algorithms used were Soundex, Double Metaphone and New York State Identification and Intelligence System. Originally they were developed to improve recall in information retrieval systems when the exact spelling of a word was unknown. Thus, they can be interpreted as broad transcription algorithms. Also, text was tran-

scribed to IPA using the Carnegie Mellon University Pronouncing Dictionary (CMU) resulting in a much narrower transcription. After transcribing, the samples were segmented into character n-grams of sizes 2–9. Then, the TF-IDF score for n-grams with a document frequency of at least 5 but not more than 5% of the training set were calculated. The scores were then used for training a linear C-Support Vector Machine. Using only features generated by the phonetic algorithms, the F_1-score was worse than using plain character n-grams. But when these features were combined with plain n-grams they increased the F_1-score. Double Metaphone and plain n-grams resulted in the largest increase of 0.56%. Also, it turned out that in all cases the broader transcriptions outperformed the narrow CMU transcription, except when using only Soundex features. Thus, a transcription that is too narrow might increase feature noise and thereby damage the classifier's performance.

Chapter 4

Datasets and Transcription

4.1 Datasets

We use learning-based classification algorithms. This means, given a set of rules, they try to induce the underlying patterns of a training set. The resulting patterns are then used to classify unseen entities of a test set. To train and test our algorithms we use two datasets, each consisting of labeled text pairs. A pair has the label *True* if both texts were authored by the same person and *False* if not.

First, we will use the small official dataset (FF) from the PAN 2020 task on Authorship Verification from Bevendorff et al. [2020a]. This allows us to compare our results to the other methods submitted. It consists of 52.601 text pairs collected from the fanfiction website `fanfiction.net`. The dataset file is formatted in the PAN 2020 format with each line containing a json object with the text pair, an ID, and optionally some additional information such as the corresponding fandoms[1]. In contrast, the large official dataset contains 256.000 samples. This is roughly five (4.86) times as many samples as the small dataset. Efforts have been made to maximally optimize the methods used, but due to certain implementation details, the utilization of the large Fanfiction dataset remains infeasible for now.

We source the second dataset (GB) from Bevendorff et al. [2019a]. It presents a dataset containing science fiction and adventure texts from the 19th and 20th century, compiled from books from Project Gutenberg[2]. As discussed earlier, the aim of this dataset is to reduce common biases in datasets for Authorship Verification. This makes it a good candidate for evaluating new authorship verifiers. With only 262 text pairs, it is much smaller than the first

[1] The franchise a fanfiction text belongs to. It can be seen as the topical domain of the text.

[2] `https://www.gutenberg.org/`

dataset used. To maximally use the information in this dataset, we employ cross-validation in our evaluation instead of a standard train-test-split method. As this dataset is in the old format, used before PAN 2020, we converted it to the new PAN 2020 format for standardization[3].

4.2 Transcription

We use a range of open-source libraries to transcribe the datasets. Figure 4.1 shows the process a given text undergoes during transcription. Because the Fanfiction dataset is at times noisy and contains artifacts that are phonetically irrelevant (e.g., long punctuation sequences, HTML-tags), we clean it with the following steps:

1. Remove tokens longer than 23 characters.
   → The longest token occuring in the Fanfiction dataset that also occurs in the ASPELL[4] dictionary is 23 characters long. Longer tokens are mainly artifacts.

2. Remove tokens with 3 or more punctuation symbols.
   → Tokens with many punctuation symbols are mainly artifacts.

3. Remove tokens containing symbols that are *not* in the following set:
   $\{symbol \mid isTranscribable(symbol) \wedge isPunctuation(symbol)\}$
   e.g. $\{a, b, c, \ldots, \tilde{n}, \ddot{e}, \ldots, !, ?, ", \ldots\}$
   → Tokens with such non-transcibable symbols do not create meaningful transcriptions.

4. Replace all double quotes with single quotes.
   → During the creation of the Fanfiction dataset all types of quotes were normalized to double quotes. This leads to combinations that are not transcribed correctly (e.g. "I"m" is erroneously transcribed to [ɪm] instead of [aɪm]). On the other hand, single quotes used in place of double quotes do not present any difficulties in transcribing.

5. Remove excessively long or short texts (< 20500 and ≥ 22500 characters, around 1.6% of the data).

The actual transcription steps are the same for the texts from both datasets. First, we transcribe a given text to *IPA* using g2pE introduced by Park and Kim [2019]. It works as follows:

[3]The code for the conversion is available at `https://github.com/av-pt/NAACL-19`
[4]`http://aspell.net/`

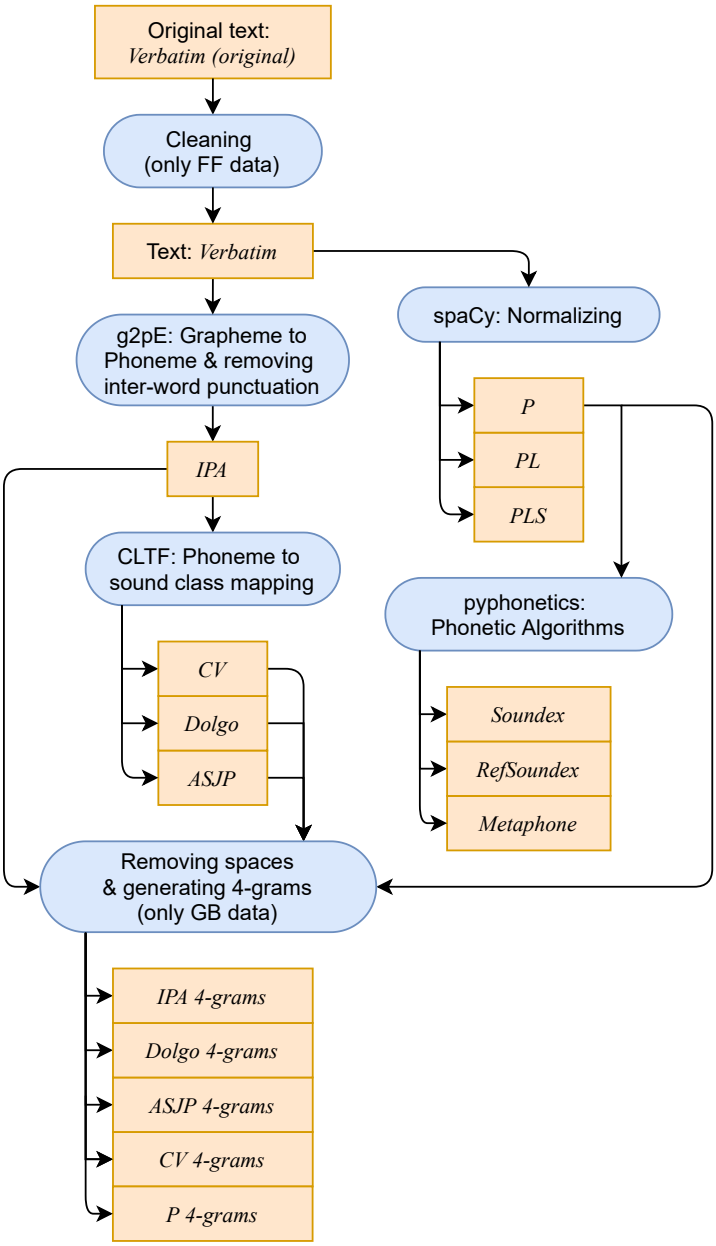

Figure 4.1: Transcription setup, orange = data, blue = process.

1. Expand numbers and currency symbols (e.g., "$400" to "four hundred dollars").

2. Use part-of-speech information to find the correct pronunciations for heteronyms, i.e., for words that have multiple context-dependent pronunciations.

3. Look up pronunciations in the Carnegie Mellon University Pronouncing Dictionary[5].

4. For out-of-vocabulary words use a neural net model to predict their correct pronunciations.

We use this method over a simpler approach because it exploits word context to find the correct pronunciation. Additionally, it creates *IPA* representations segmented into phonemes. This is important for the next step, generating the broader sound class transcriptions using the Cross-Linguistic Transcription Systems project by Anderson et al. [2018]. CLTS serves as a phoneme-by-phoneme mapping between different transcription and sound class systems. Given a list of *IPA* transcribed phonemes, they can be mapped to a range of other systems. As words in *IPA* can contain arbitrary supra-segmental letters, and it is hard to segment these words into phonemes after transcribing, List et al. [2018] recommends using segmented IPA representations. Transcribing, for example, the word "make" to *IPA* results in [meɪk]. In contrast to other algorithms, g2pE produces the correct segmentation [m eɪ k]. Using CLTS to convert this to the *Dolgo* system we correctly get "MVK". If we were, for example, to naively segment [meɪk] to [m e ɪ k] by interpreting each *IPA* symbol as a phoneme, we would incorrectly get "MVVK" as a result for the *Dolgo* system. For the Gutenberg dataset, we also generate space separated character 4-grams for the systems above.

Punctuation and stop word removal, as well as lemmatization is done with spaCy[6] for speed and robustness. For the punctuation-removed data (P) we also create 4-grams. They can be used as a generic n-gram approach compared to n-gram approaches using transcriptions as the transcriptions above also have punctuation removed. The other phonetic algorithms — *Soundex*, *RefSoundex* and *Metaphone* — work with verbatim text on word level, i.e., they do not use context but transcribe each word in isolation. Thus, we space-tokenize the punctuation-removed data (P) and use the resulting lists as input for these algorithms. For the transcriptions themselves we use the library pyphonetics[7].

[5] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict/`

[6] `https://spacy.io/`

[7] `https://github.com/Lilykos/pyphonetics`

The source code in this library is based on Talisman.js[8] which itself is based on the Apache commons codec[9]. The source code for transcribing datasets formatted in the PAN 2020 standard is available on GitHub[10].

4.3 Transcription Characteristics

To better understand the characteristics of the phonetic transcription systems and the idiosyncrasies of the datasets, we conduct some preliminary investigations. We calculate the vocabulary size scaling factor (*VSSF*) for each transcription system by determining the ratio of the number of distinct lexical types before and after transcribing. A *VSSF* of 1.5, for example, indicates that the examined transcription system increases the vocabulary size by 50%. This way, we can assess the granularity of the different transcription systems. A vocabulary reduction of 50%, for example, indicates that on average two words are binned into one transcription. In practice, there may be some transcriptions grouping many words while many transcriptions would have a one-to-one mapping to a single word. We calculate the absolute vocabulary size and the *VSSF* per transcription system per dataset. Table 4.1 shows the results.

First, let us take a look at findings for the Gutenberg dataset, which are visualized in Figure 4.2. The *verbatim* text contains 50277 types. Both *IPA* and *ASJP* increase the vocabulary size by a significant amount, 20.68% and 11.21% respectively. This is to be expected as the alphabet of both systems is larger than the alphabet of *verbatim* text and thus more types can be generated. The text with removed punctuation (*P*) retains the same amount of types as *verbatim* text because punctuation symbols are not counted towards the vocabulary size. By further lemmatizing the texts (*PL*), more tokens are binned and the resulting vocabulary is reduced by 18.6%. Eliminating stop words (*PLS*) removes 220 more types. The *Dolgo* system has an even smaller amount of types, but still retains more type granularity than the more simple phonetic algorithms. *RefSoundex* and *Metaphone* reduce the vocabulary size by 41.44% and 47.29% respectively. Because of its length restriction to four characters, *Soundex* can at most produce 8,918 unique types (A000 to Z666) with only 4250 of them appearing in the data. Lastly, *CV* reduces the number of types the most and retains only 1954 types. A reduction to 3.89% of the original vocabulary size implies that on average ca. 26 words are binned into one transcription.

[8] `https://yomguithereal.github.io/talisman/`

[9] `http://commons.apache.org/proper/commons-codec/userguide.html`

[10] `https://github.com/av-pt/ba-util`

Table 4.1: Effects of transcription systems on the vocabulary sizes of the datasets. *Verbatim* represents plain English text. *Verbatim (original)* is the uncleaned version of the Fanfiction dataset.

System	GB absolute	GB VSSF	FF absolute	FF VSSF
Verbatim (orig.)	–	–	795621	1.0426
IPA	60673	1.2068	782424	1.0253
ASJP	55913	1.1121	691146	0.9057
Verbatim	50277	1.0	763097	1.0
P	50277	1.0	754293	0.9885
PL	40924	0.814	739629	0.9692
PLS	40704	0.8096	739530	0.9691
Dolgo	36288	0.7218	384440	0.5038
RefSoundex	29441	0.5856	229360	0.3006
Metaphone	26501	0.5271	210973	0.2765
Soundex	4250	0.0845	6471	0.0085
CV	1954	0.0389	9436	0.0124
IPA 4-grams	176092	3.5024	–	–
P 4-grams	103983	2.0682	–	–
ASJP 4-grams	78246	1.5563	–	–
Dolgo 4-grams	4061	0.0808	–	–
CV 4-grams	16	0.0003	–	–

Figure 4.3 shows the results for the same analysis but using the Fanfiction dataset. Both plots are predominantly similar but exhibit a few interesting differences stemming from the characteristics of the transcription systems and the datasets. Note that the Fanfiction dataset is substantially larger than the Gutenberg dataset, also leading to a larger total vocabulary count.

First, the vocabulary size of the uncleaned (original) *verbatim* text is 4.26% larger than that of the cleaned one. This was expected as we removed certain words during cleaning. Next, it can be observed that there is an unexpected difference between *verbatim* text and punctuation-removed text (P). This may stem from the Fanfiction dataset including many more different punctuation symbols which are removed to create the P transcription but not when counting types in *verbatim* text. Also, the relative difference between P on the one hand and *PL* as well as *PLS* on the other is much smaller. This could indicate that the Fanfiction dataset has many out-of-vocabulary tokens that are not easily lemmatized. Compared to the Gutenberg dataset, which consists of texts from published books, this would make sense as the acceptance criteria

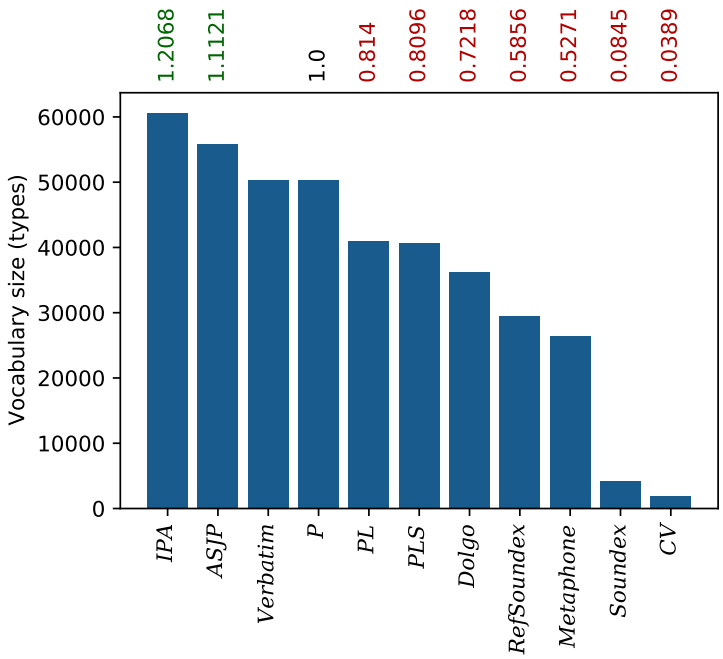

Figure 4.2: Vocabulary sizes for transcriptions on Gutenberg dataset with *VSSF*s above.

for Fanfiction stories are much lower than those for books. For the phonetic algorithms, *Soundex* has a smaller vocabulary than *CV*. As mentioned above, the codes generated by the *Soundex* algorithm are bound to 4 characters in length. Because *CV* tokens are only restricted to a binary alphabet, but do not have any length restrictions, with a large enough text sample the *CV* vocabulary outnumbers the *Soundex* vocabulary. To substantiate this claim, we created a vocabulary list for each text pair in the Fanfiction dataset. Then we accumulated the vocabulary lists one by one to examine if the transcribed texts follow Heaps' Law as defined in Schütze et al. [2008]. Figure 4.4 shows the sizes for the accumulated vocabulary for *Soundex* and *CV* when reading in the texts from the Fanfiction dataset in a shuffled order. The vocabulary of the *Soundex* transcription grows fast in the beginning but then begins to max out at around 6000 types. For the *CV* transcription on the other hand, the accumulated vocabulary size grows slowly in the beginning, probably due to its restricted alphabet, but does not hit a ceiling and continues to grow beyond *Soundex*'s vocabulary size.

Arguably, the most notable difference is that when transcribing the Gutenberg dataset *ASJP* leads to an increase in vocabulary size whereas using the Fanfiction dataset *ASJP* surprisingly results in a significant reduction of the

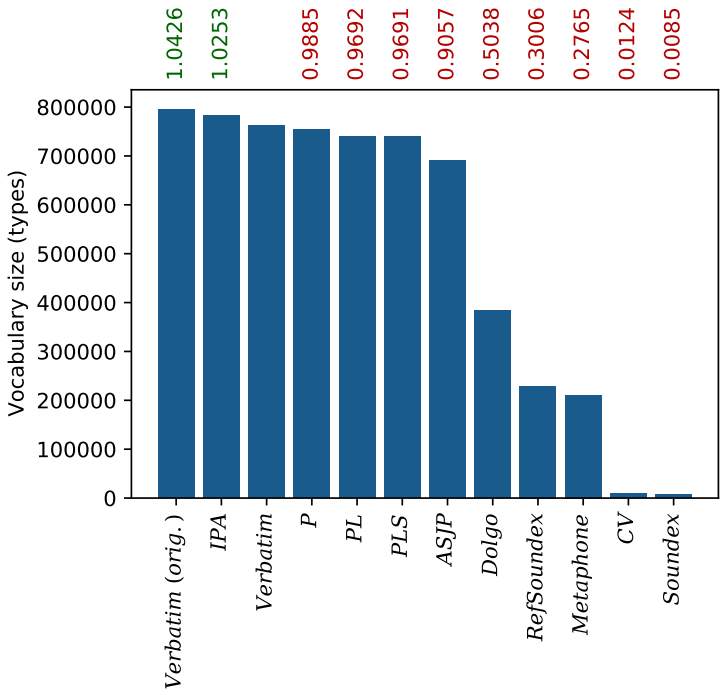

Figure 4.3: Vocabulary sizes for transcriptions on Fanfiction dataset with *VSSF*s above.

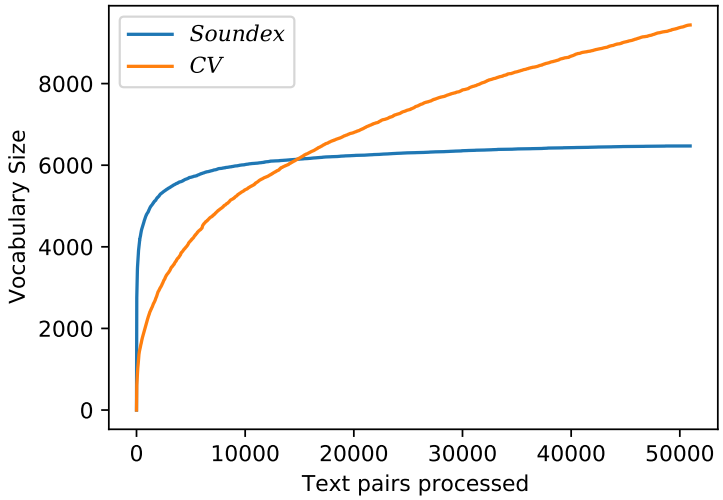

Figure 4.4: Accumulated vocabulary size for *Soundex* and *CV*, shuffled Fanfiction dataset.

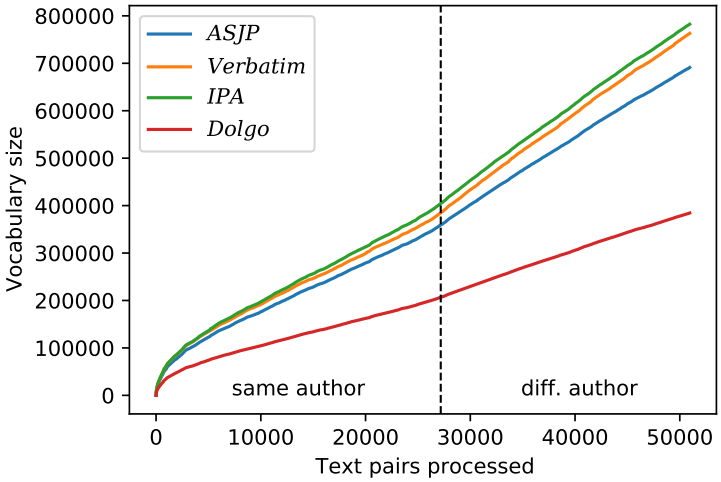

Figure 4.5: Accumulated vocabulary size for *Verbatim*, *IPA*, *ASJP*, and *Dolgo*, in-order Fanfiction dataset.

vocabulary. Note that as shown in Figure 4.1, *ASJP* results from the *IPA* transcription. To attain a clearer view of what is happening here, we also calculate the accumulated vocabulary sizes for *Verbatim*, *IPA*, *ASJP*, and *Dolgo*, shown in Figure 4.5. This plot poses two additional questions[11]:

1. Why is there a sudden change in slope in the accumulated vocabulary size?

2. Why do *IPA* and *verbatim* text diverge on the left side but stay at a constant difference after the change in slope on the right side?

The first question has an obvious answer: The Fanfiction dataset is sorted. The first half consists of only same-author pairs with all different-author pairs residing in the second half. Table 4.2 shows information on the author distribution of both datasets segmented into same-author and different-author parts. With 47813 authors, the different-author part of the Fanfiction dataset has around 7.47 times as many authors as the same-author part. For the same-author part, one author wrote 1.036 individual texts on average whereas for the different-author part this number is 8.7022[12]. It comes as no surprise then that — despite individual text and vocabulary sizes being nearly equal for both parts — the vocabulary of the different-author part is much more diverse. This

[11] Both phenomena persist when using single texts instead of text pairs for the cumulative vocabulary size analysis.

[12] Or in terms of dataset samples, on average authors contributed to 0.518 and 4.3511 text pairs respectively.

Table 4.2: Author distribution of the datasets used.

	Gutenberg	**Fanfiction**
No. of authors in same-author part	54	6397
No. of authors in different-author part	56	47813
No. of authors in both parts	53	1555
Texts per author in same-author part	4.8148	8.7022
Texts per author in diff-author part	4.7143	1.036

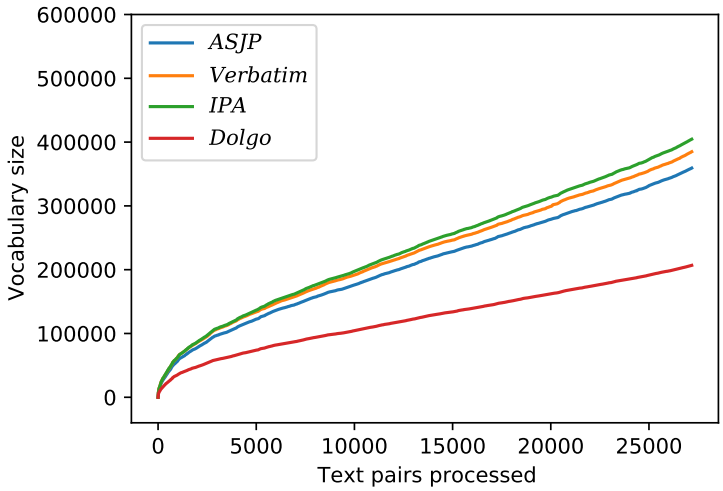

Figure 4.6: Same-author part of accumulated vocabulary size for *Verbatim*, *IPA*, *ASJP*, and *Dolgo*, in-order Fanfiction dataset.

diversity leads to the higher slope in the different-author part. As a side note, the bias-mitigated Gutenberg dataset does not exhibit a change of slope when analyzing it as above. This is also reflected in the number of authors for the same- and different-author parts in table 4.2. For the Gutenberg dataset, the number of authors for both parts are almost identical, and most authors appear in both parts. Also, the number of texts an author contributed to the dataset is nearly equal between both parts. We conclude that the number of authors is correlated to the vocabulary size of a given dataset. Further investigations are necessary to determine whether the difference in author distribution and thus in vocabulary size in the Fanfiction dataset has an effect on the results from the PAN 2020 task where this dataset was used.

As of yet, we do not have an answer to the second question. To change the perspective of this phenomenon, Figures 4.6 and 4.7 show the cumulative vocabulary size for the same-author and different-author part in isolation. As expected, all curves follow Heaps' Law and the total vocabulary of the same-

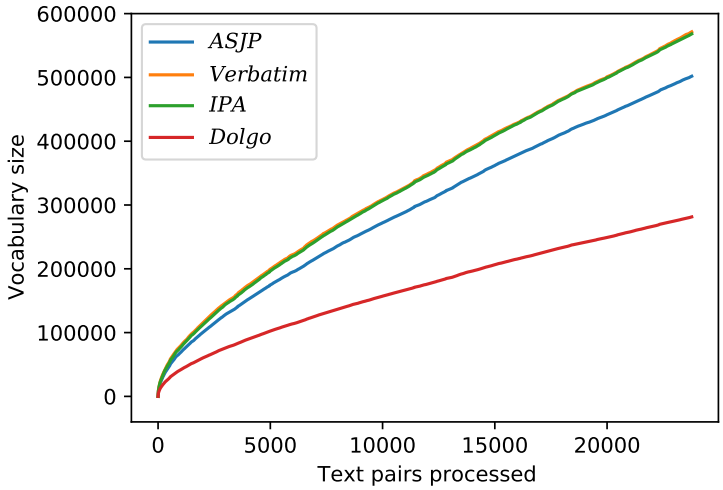

Figure 4.7: Different-author part of accumulated vocabulary size for *Verbatim*, *IPA*, *ASJP*, and *Dolgo*, in-order Fanfiction dataset.

author part is lower than that of the different-author part. The surprising difference is that the curves for *Verbatim* and *IPA* in Figure 4.7 do not diverge. In contrast to the same-author part, for the different-author part the transcription to *IPA* does *not* lead to an increased vocabulary size. We investigated some possible explanations of this phenomenon. The samples from both parts are transcribed in one coherent process and also plotted in one go, lowering the probability of an implementation error. The percentage of alphabet characters (a-z, A-Z) in both parts is almost equal. The only small difference we found is in the fraction of words that occur in the ASPELL dictionary. For the same-author part 15% of the words are in ASPELL while for the different-author part only 10% are in ASPELL.

We suspect that the decrease in vocabulary size of *ASJP* compared to *verbatim* text (Fig. 4.3) has two reasons. First, the missing increase for the vocabulary size of *IPA* in the different-author part. As *IPA* is the transcription preceding *ASJP*, the vocabulary size of the former directly impacts the that of the latter. Second, in Figure 4.6 we observe that *ASJP* still produces a lower vocabulary count, even for the same-author part only. This lets us speculate that some other factor, e.g., the quality of the dataset, must also play a roll in its transcription. Maybe both, the same-author and the different-author part, are affected by the same phenomenon with the same-author part affected only slightly. This hypothesis could be supported by comparing the vocabulary increase of *IPA* between both datasets. With the Gutenberg dataset, *IPA* increases the vocabulary by 20.68% (Fig. 4.2) while with the Fanfiction dataset the vocabulary is only increased by 2.53% (Fig. 4.3). Still, this comparison

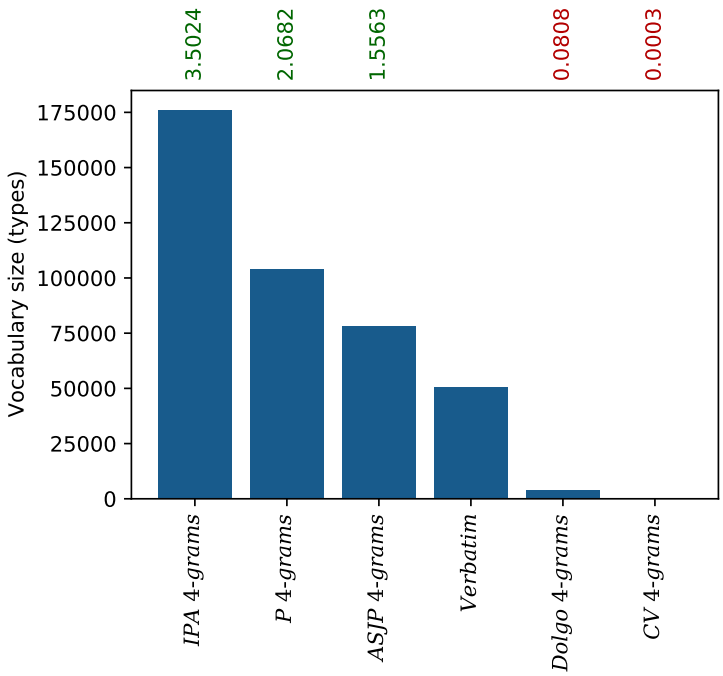

Figure 4.8: Vocabulary sizes for 4-grams compared to *Verbatim*.

should be interpreted cautiously because of the bespoken size differences of the datasets. To this end, further investigation is needed.

Figure 4.8 shows the vocabulary sizes for the 4-grams, i.e., the amount of unique 4-grams generated from each transcription of the Gutenberg dataset, compared to the vocabulary size of *verbatim* text. With a binary alphabet, *CV* creates only 16 unique 4-grams. This is followed by *Dolgo*, which could at most create 20736 4-grams of which 4061 appear in the data. *ASJP* increases upon the vocabulary count of *verbatim* text, as it has a bigger alphabet. The 4-grams generated from punctuation-removed text have an even higher count. This is due to intra-word punctuation not being removed and thus retaining many of the punctuation symbols from *verbatim* text. Lastly, *IPA* generates by far the most 4-grams, as it has the biggest alphabet of all transcriptions used.

Chapter 5

Experiments

As shown in Figure 5.1, the main experimental setup can be structured into four parts. In a preprocessing step, we standardize the Gutenberg dataset to the new PAN 2020 format and we clean the Fanfiction dataset. Then, we transcribe the datasets using the phonetic transcription methods defined earlier. The resulting datasets, as well as the original dataset, are then used as the inputs to two widely used Authorship Verification algorithms which are described in more detail in this chapter. For all experiments, we conduct three cross-validation runs with ten folds each. We then average the results and compute the Bonferroni-corrected[1] statistical significance using a paired t-test as the test statistic. Finally, we analyze the results.

5.1 Compression Approach

The first approach, originally from Teahan and Harper [2003] and later adapted to Authorship Verification and employed as a benchmark in PAN 2020, uses a text compression method to determine the chance that two texts were written by the same author. Text compression can be seen as encoding a given text with an encoding that is optimized for this text. As discussed in Brown et al. [1992], by determining this encoding, text compression can be used to estimate an upper bound to the entropy, i.e., the amount of information of characters in English text. More specifically, by using the compression model optimized on some text A, the cross-entropy of encoding a text B with this model can be calculated. During training, this is done for each pair in both directions. The mean and average of the distance between the resulting cross-entropies are then used to train a logistic regression model. The smaller the resulting

[1] As we conduct 30 runs on the same data, the likelihood of encountering a rare configuration that performs well and accepting it as statistically significant is high. Thus we divide the p-values for accepting statistical significance by 30.

difference, the more similar the texts, and the higher the chance that both are written by the same author. The compression model used is Prediction by Partial Matching (PPM), a standard algorithm for lossless text compression, first introduced by Cleary and Witten [1984]. As done in PAN 2020, we employ an uncertainty interval with a radius of 5%, i.e., predictions that are in the interval [0.45, 0.55] are given a *non-answer* classification. The source code used is based on a reimplementation of the Authorship Attribution approach from Teahan and Harper [2003] as part of a reproducibility study in Potthast et al. [2016]. The adaption for Authorship Verification stems from PAN 2020[2]. The source code extending the algorithm to use phonetic features and adding cross-validation functionality is available on GitHub[3].

5.2 Unmasking Approach

Unmasking was first introduced by Koppel and Schler in 2004. In short, it exploits the degradation of classifier accuracy when removing distinguishing features. It turns out that iteratively removing those features leads to a faster degradation on text pairs by one author than on those by different authors. Thus, the algorithm "unmasks" the text pairs and thereby reveals the information needed for classification.

This approach comprises two steps: First, a cross-validation method is employed to create the accuracy degradation curves for all training samples. Secondly, a meta-classifier is trained on the resulting curves to differentiate between same-author and different-author curves.

To compute a curve for a pair, both texts are chunked into parts longer than 500 words without splitting paragraphs. The 250 words with highest average frequencies in the two texts are used as features. In a 10-fold cross-validation, linear support vector machine (SVM) models are trained to classify if a chunk belongs to the first or the second text. The resulting accuracy is noted and the three most discriminating positive and negative features are removed from the feature set. The cross-validation and feature removal are repeated until no features are left. The set of curves is then used to train a linear SVM model as a meta-classifier. As brought to the point by Bevendorff et al. [2019b], the features used are "the curve points, the curves' point-wise first- and second-order derivatives, and the derivatives sorted by steepest point-wise drop".

Unmasking is one of the most robust Authorship Verification algorithms. But as it requires sufficient chunks of no less than 500 words in length, it is only

[2] `https://github.com/pan-webis-de/pan-code/tree/master/clef20/authorship-verification`

[3] `https://github.com/av-pt/teahan03-phonetic`

applicable for book-length texts. To counter this, Bevendorff et al. [2019b] generalizes the algorithm to accommodate for short texts. Chunks are generated by oversampling the bag-of-words pool of a given text. Words from this pool are picked randomly without replacement until a length of 700 words is reached and the pool is reset afterwards. In total, 30 chunks are generated which are then used for curve generation as above with the only exception that the five most positive and negative features are removed instead of only three. As this approach introduces a significant amount of variance in the resulting curves, the Unmasking step is repeated multiple times and the curves are averaged. It is recommended to average at least 15–20 Unmasking runs for each text pair. In our experiments, we average the curves of 32 Unmasking runs per text pair with varying chunk sizes of 500, 600, and 700 words respectively. In the implementation supplied by Bevendorff et al. [2019b][4] and used as the basis for our research, the meta-classifier uses the curves' "central-difference gradients (first- and second-order), as well as their gradients sorted by magnitude" as features. Also, the implementation predicts labels instead of confidence values and hence does not produce *non-answers*. The source code extending the Unmasking framework for the use of phonetically transcribed datasets and implementing a cross-validation functionality is available on GitHub[5]. Note that we only use the Gutenberg dataset for Unmasking as processing the larger Fanfiction dataset for each transcription proved to be too time-consuming.

[4] `https://github.com/webis-de/unmasking`
[5] `https://github.com/av-pt/unmasking`

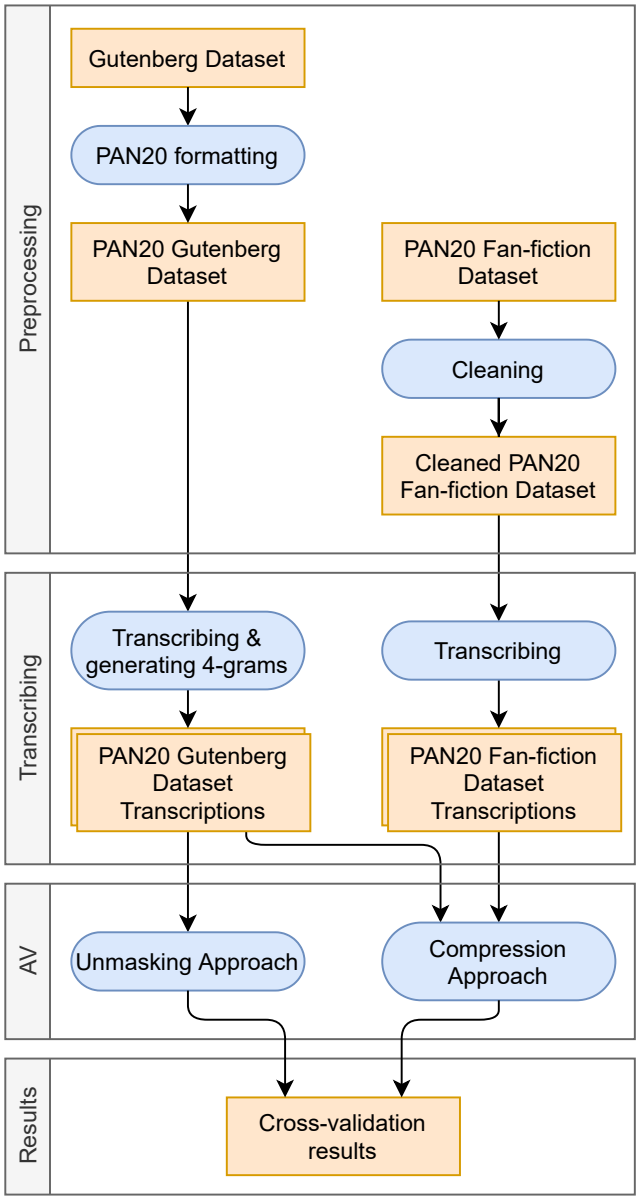

Figure 5.1: Experimental setup, orange = data, blue = process.

Chapter 6

Results

For the evaluation of our approaches we use several traditional as well as recently proposed measures. We source the definitions for this section from Schütze et al. [2008]. We use the convention that a pair is in the positive class iff both texts are written by the same author. tp, tn, fp, fn stand for the number of cases that where classified correctly as positive (true positives), correctly as negative (true negatives), falsely as positive (false positives), and falsely as negative (false negatives) respectively.
The **precision** of a classifier is the percentage of correct positive classifications tp over all classifications $tp + fp$:

$$pre = \frac{tp}{tp + fp}$$

Thus, a precision approaching 1 indicates that an AV classifier's same-author predictions are near fully correct, meaning that there are nearly no false positives.
The **recall** of a classifier is the percentage of correctly classified positive samples tp over all positive samples $tp + fn$:

$$rec = \frac{tp}{tp + fn}$$

The lower the recall, the fewer same-author cases are recognized and predicted as such by an AV classifier. In turn, a recall of 1 indicates that all same-author cases have been correctly identified.

Ideally, we want a system that classifies all same-author cases and only those as positive. To measure this behavior, the **F1-score** can be used:

$$F_1 = 2 \cdot \frac{prec \cdot rec}{prec + rec}.$$

If both, precision and recall, approach 1 the F_1-score of an AV classifier also approaches its maximum of 1. Note that the F_1-score weights precision and recall equally. Especially for forensic Authorship Verification applications though, a high precision is more important than a high recall, as same-author decisions might be used as evidence and therefore must be reliable. Also, in our setup, the F_1-score ignores true negatives and therefore does not give an insight into how well the classifier detects different-author cases correctly. For this, the different-author class would need to be assigned the positive label.

To mitigate some of the problems of the measures above and to better assess AV classifier performance, we use two more recently introduced measures. To include same-author and different-author classifications in the evaluation, one could use the accuracy:

$$acc = \frac{tp + tn}{n}$$

where $n = tp+tn+fp+fn$ is the total number of cases. However, as Bevendorff et al. [2019b] points out, the results are often uncertain. Also, in real-world applications wrong answers might be worse than *non-answers*. Therefore, to give classification systems the option to withhold answers for difficult-to-decide cases, we use the **c@1-score** introduced by Peñas and Rodrigo [2011] and adopted by PAN:

$$c@1 = acc + \frac{acc}{n} \cdot n_u$$

where n is again the total number of cases, and n_u is the number of undecided cases. This way, undecided cases count towards the c@1-score as if they were answered with the accuracy of the decided cases. When an AV classifier gives an answer to all cases, the c@1-score is equivalent to the accuracy of the classifier. A system that leaves all cases unanswered receives a score of 0.

Lastly, we use the **F0.5u-score** introduced by Bevendorff et al. [2019b]:

$$F_{0.5u} = \frac{(1 + 0.5^2) \cdot n_{tp}}{(1 + 0.5^2) \cdot n_{tp} + 0.5^2 \cdot (n_{fn} + n_u) + n_{fp}}.$$

As mentioned above, a high precision result is more reliable than a high recall one. To take this into consideration, the $F_{0.5u}$-score weights precision two times as much as recall. In addition, it also allows the classifiers to give *non-answers*. However, as unanswered cases are often not useful in real-world applications, it interprets them as wrong answers. Thus, the $F_{0.5u}$-score emphasizes on the precision of an AV classifier.

The results of our experiments can be seen in Tables 6.1, 6.2, and 6.3. The tables show the absolute values of the results for the transcription systems

Table 6.1: Results for Unmasking using the Gutenberg dataset with Bonferroni-corrected significance markers.

System	Precision	Recall	F1	F0.5u (F0.5)	c@1 (Accuracy)
Verbatim	0.7545	0.6539	0.6977	0.6784	0.6875
IPA	0.7537	0.6701	0.7004	0.6641	0.6833
ASJP	0.7284	0.667	0.6893	0.6455	0.676
Dolgo	0.7551	0.6733	0.7046	0.6617	0.6792
RefSoundex	0.7297	0.6275	0.6673	0.6233^{**}_{-}	0.6542
Metaphone	0.7193	0.6028	0.648^{*}_{-}	0.6143^{**}_{-}	0.6423^{**}_{-}
Soundex	0.7333	0.6363	0.6677	0.6234^{**}_{-}	0.6512
CV	0.6316^{**}_{-}	0.5757	0.5936^{**}_{-}	0.5171^{***}_{-}	0.5381^{***}_{-}
IPA 4-*grams*	0.6536^{**}_{-}	0.5814	0.6043^{**}_{-}	0.5505^{***}_{-}	0.5976^{***}_{-}
P 4-*grams*	0.718	0.6339	0.6658	0.6207^{*}_{-}	0.6387^{*}_{-}
ASJP 4-*grams*	0.721	0.6148	0.6553	0.6086	0.6257
Dolgo 4-*grams*	0.6894	0.5964	0.6245	0.5558^{***}_{-}	0.6012^{**}_{-}
CV 4-*grams*	0.6161^{**}_{-}	0.5853	0.5926^{**}_{-}	0.4868^{***}_{-}	0.5151^{***}_{-}
P	0.7155	0.6672	0.6859	0.6452	0.6745
PL	**0.7707**	**0.6801**	**0.7153**	**0.6876**	**0.7079**
PLS	0.7676	0.6614	0.7031	0.6572	0.6905

employed and their statistical significance compared to the results of *verbatim* text in the first row.[1]

First, we will discuss the results of the Unmasking approach using the Gutenberg dataset. Before analyzing the final results, let us take a look at the sets of the degradation curves generated through Unmasking. The curve sets generated for each transcription can be split into five types depending on how fast the accuracy drops. For *Verbatim*, *IPA*, *ASJP*, *Dolgo*, *RefSoundex*, *Metaphone*, *Soundex*, *P*, and *PL*, the curves degrade over the entirety of the iterations, giving the maximal resolution for further use as features. Figure 6.2 shows this "best-case" resolution for *Verbatim*. Next, the curves for *IPA* 4-*grams*, *P* 4-*grams*, *ASJP* 4-*grams*, and *Dolgo* 4-*grams* degrade quicker as can be seen in Figure 6.3 for *IPA* 4-*grams*. This may be caused by a reduction of author-distinguishing features through 4-gram-generation. We suspect that —

[1] $*$: $p < 0.05$; $**$: $p < 0.01$; $***$: $p < 0.001$; $+$ marks an increase and $-$ marks a decrease compared to the top row; bold text marks the highest value of each column.
E.g., 0.5936^{**}_{-} indicates a decrease of the current transcription's performance compared to *verbatim* text with $p < 0.01$.

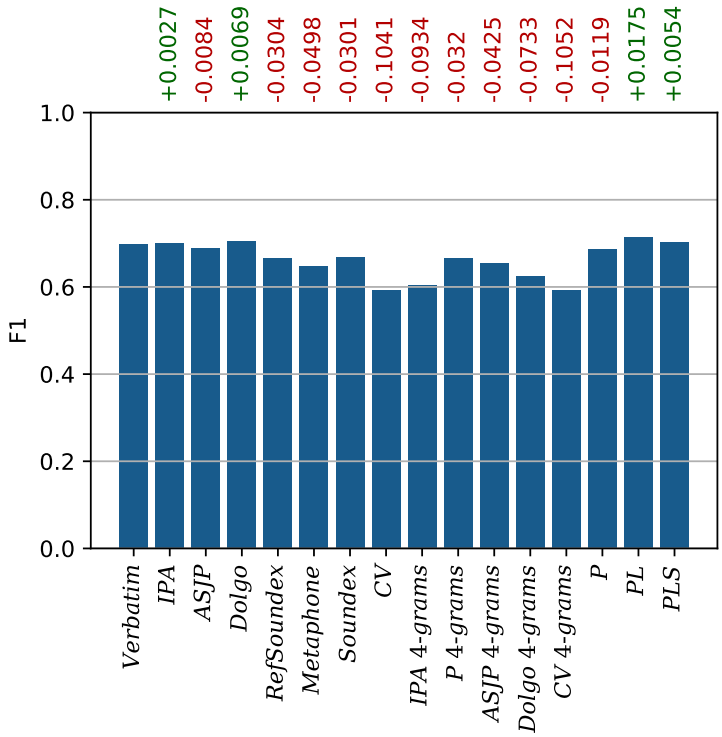

Figure 6.1: F_1-score and differences for Unmasking using the Gutenberg dataset, significant changes: *Metaphone* 4-*grams***, *IPA* 4-*grams***, and *CV* 4-*grams***.

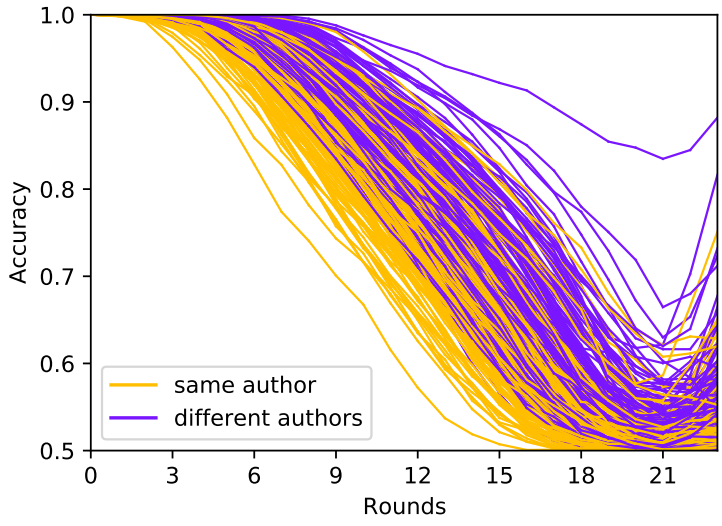

Figure 6.2: Unmasking curves for *Verbatim* using the Gutenberg dataset.

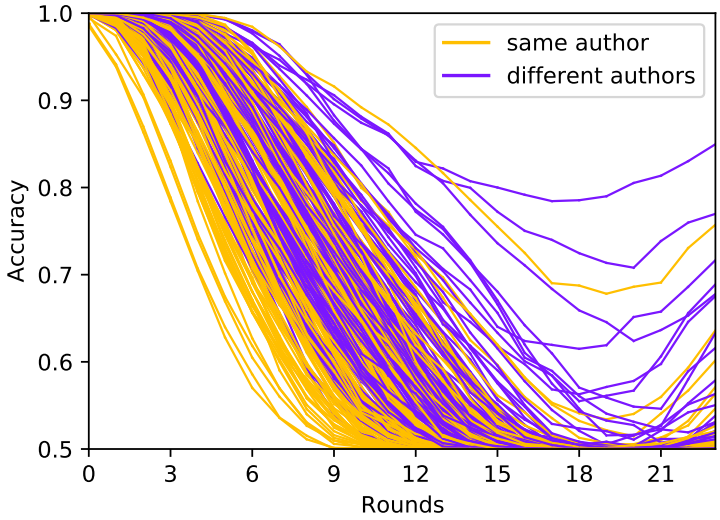

Figure 6.3: Unmasking curves for *IPA 4-grams* using the Gutenberg dataset.

analogous to stop words as discussed in Chapter 3 — certain of the possible 4-grams appear often and are suited well for distinguishing between authors while the less frequent ones are not, resulting in a faster curve degradation. Further investigation is needed to confirm or deny this hypothesis. For *CV 4-grams* (Fig. 6.4) the accuracy drops to zero within a few Unmasking iterations, for the simple reason that it only contains 16 different types that are used as features. The standard *CV* transcription exhibits more chaotic curves, as shown in Figure 6.5. Apparently, a binary alphabet does not give the linear SVMs enough information to make robust guesses during curve generation. Lastly, for *PLS* (Fig. 6.6) the additional stop word removal leads to a much slower accuracy degradation. As discussed earlier, stopwords are good features for distinguishing between authors. Removing these common words results in the remaining features being more topic-related. Thus, the given texts can be distinguished more easily even after removing certain of the remaining features. As our goal, however, is not to retain a high accuracy of the curves, but to increase the differences between same-author and different-author curves, omitting stop words does not seem promising.

The results of the cross-validation for Unmasking using the Gutenberg dataset can be seen in Table 6.1. As mentioned in Section 5.2, Unmasking does not produce *non-answers* and thus $F_{0.5u}$[2] is reduced to $F_{0.5}$ and $c@1$ is equal to the classifier's accuracy. The results of the individual folds have a high variance, characteristic of the probabilistic nature of Unmasking. This is

[2]Weighting precision two times as much as recall, but not accounting for *non-answers*.

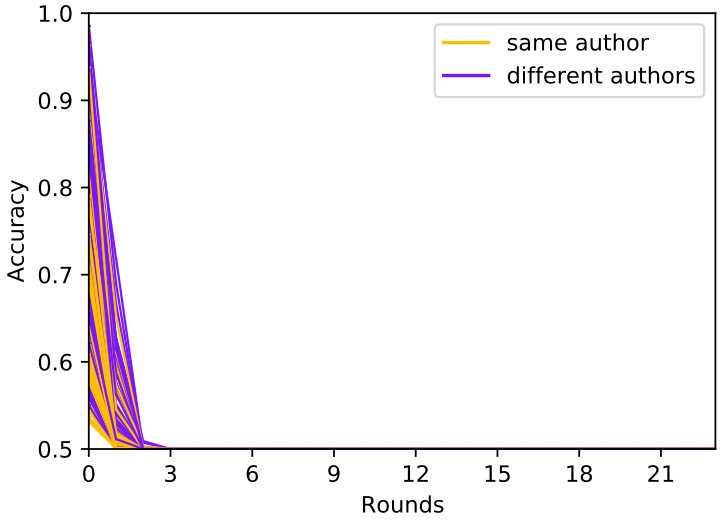

Figure 6.4: Unmasking curves for *CV* 4-*grams* using the Gutenberg dataset.

reflected in the results as the smallest change accepted as statistically significant ($p < 0.05$) is the decrease of $c@1$ for *P* 4-*grams* of 4.88% from 0.6875 to 0.6387. Generally, all statistically significant changes result in a reduction of performance with *CV*, *CV* 4-*grams*, and *IPA* 4-*grams* reducing the performance the most overall. As already discussed above, the Unmasking curves for *CV* and *CV* 4-*grams* exhibit characteristics suggesting a decrease in performance as the linear SVMs employed in the meta-classification do not have enough information to produce meaningful predictions. For *IPA* 4-*grams* the increase in vocabulary size to 350% its original size probably results in too many features, also decreasing classifier performance. To acquire a sense for the range of performance reduction, Figure 6.1 shows the F_1-score achieved by each of the transcription systems. A slight trend can be observed regarding the granularity of the transcription systems and their F_1-score: The broader the system, the worse it performs. In general, 4-grams seem to perform worse then the transcriptions they were generated from. Still, these observations have to be taken cautiously as most of the changes are not statistically significant. Note that, although not also statistically significant, *PL* increases all scores the most, suggesting that punctuation removal and lemmatization are a more promising pre-processing step.

The results of the compression approach using the Fanfiction dataset provide a different picture. As the cleaning step of the Fanfiction dataset is phonetically-informed, we use the original data for comparison. As can be seen in Table 6.2, almost all of the changes are statistically significant. This is a

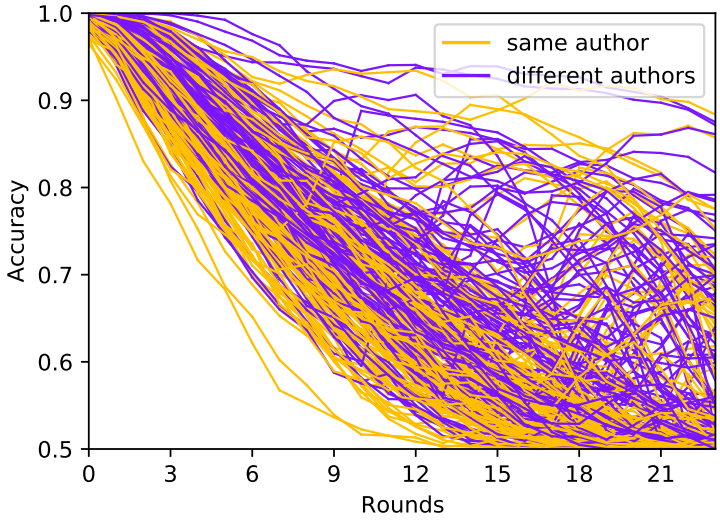

Figure 6.5: Unmasking curves for *CV* using the Gutenberg dataset.

result of the compression approach being deterministic and all variations being introduced by different configurations of cross-validation folds. The practical significances of the results, however, are low. Again, almost all experiments result in a decreased perfomance of the respective score. Surprisingly, *Metaphone* breaks this rule, yielding slight increases over *verbatim* text. Its classification precision improves upon the baseline by 0.85% leading to an improvement in $F_{0.5u}$ of 1.26%. *CV* results in a minor improvement of 0.8% in recall, but a major loss of 12.64% in precision. Figure 6.7 shows the F_1-score for this experiment. The trend of broader transcriptions leading to worse performance can be observed here as well, with the highest drop of 6.96% in F_1 coming from *CV*.

As mentioned earlier, we use the default uncertainty interval of [0.45, 0.55]. Increasing this interval improves precision and recall, because *non-answers* are omitted in their calculation and thus only those answers are counted that the classifier is more certain of. Also, $F_{0.5u}$ deteriorates as it counts *non-answers* as wrong classifications. Decreasing the uncertainty interval, on the other hand, leads to a general decrease in performance as uncertain answers are binarized and counted with the same weight as certain answers.

Lastly, Table 6.1 shows the results for the compression approach using the Gutenberg dataset. During the cross-validation for this experiment, regardless of the number of splits, the classifier returned only negative classifications for some folds of the following transcriptions: *Dolgo*, *RefSoundex*, *Soundex*, *CV*, *ASJP* 4-*grams*, *P* 4-*grams*,*Dolgo* 4-*grams*, and *CV* 4-*grams*. We suspect

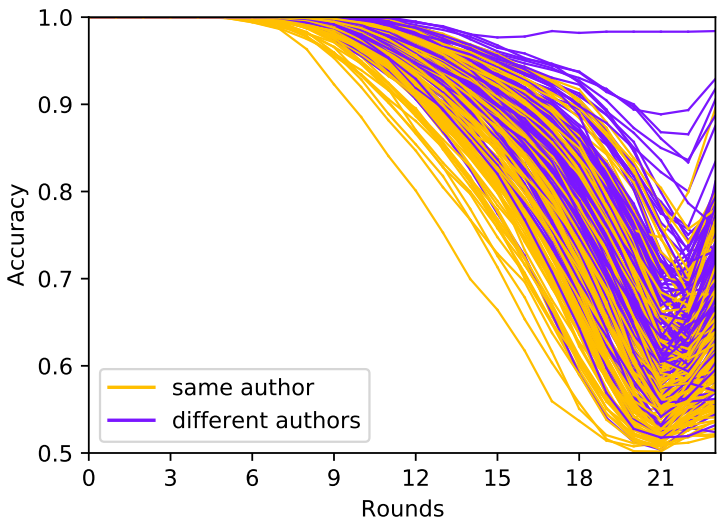

Figure 6.6: Unmasking curves for *PLS* using the Gutenberg dataset.

Table 6.2: Real caption[3]

System	Precision	Recall	F1	F0.5u	c@1
Verbatim (orig.)	0.7635	0.8092	**0.7856**	0.714	0.7431
IPA	0.7532^{***}_{-}	0.7846^{***}_{-}	0.7686^{***}_{-}	0.7098^{***}_{-}	0.7291^{***}_{-}
Verbatim	0.7604^{***}_{-}	0.8078^{**}_{-}	0.7833^{***}_{-}	0.7101^{***}_{-}	0.739^{***}_{-}
ASJP	0.7602^{**}_{-}	0.7857^{***}_{-}	0.7727^{***}_{-}	0.7148	0.7353^{***}_{-}
Dolgo	0.7474^{***}_{-}	0.7757^{***}_{-}	0.7612^{***}_{-}	0.6992^{***}_{-}	0.7174^{***}_{-}
RefSoundex	0.7564^{***}_{-}	0.7811^{***}_{-}	0.7685^{***}_{-}	0.7049^{***}_{-}	0.7259^{***}_{-}
Metaphone	$\mathbf{0.772}^{***}_{+}$	0.7907^{***}_{-}	0.7813^{***}_{-}	$\mathbf{0.7266}^{***}_{+}$	$\mathbf{0.7477}^{***}_{+}$
Soundex	0.7129^{***}_{-}	0.7717^{***}_{-}	0.7411^{***}_{-}	0.6758^{***}_{-}	0.6839^{***}_{-}
CV	0.6371^{***}_{-}	$\mathbf{0.8172}^{**}_{+}$	0.716^{***}_{-}	0.6253^{***}_{-}	0.5842^{***}_{-}
P	0.7528^{***}_{-}	0.7914^{***}_{-}	0.7716^{***}_{-}	0.7092^{***}_{-}	0.7304^{***}_{-}
PL	0.7228^{***}_{-}	0.7868^{***}_{-}	0.7534^{***}_{-}	0.6782^{***}_{-}	0.6949^{***}_{-}
PLS	0.6994^{***}_{-}	0.7773^{***}_{-}	0.7363^{***}_{-}	0.6604^{***}_{-}	0.668^{***}_{-}

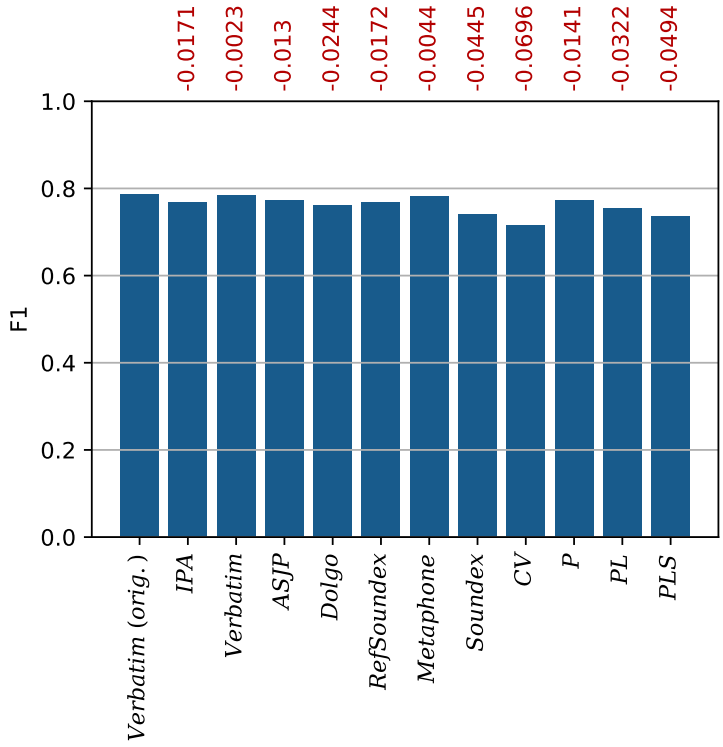

Figure 6.7: F_1-score and differences for the compression approach using the Fanfiction dataset, all changes are statistically significant.

Table 6.3: Results for the compression approach using the Gutenberg dataset compared to *verbatim* text with Bonferroni-corrected significance markers.

System	Precision	Recall	F1	F0.5u	c@1
Verbatim	0.871	**0.7354**	0.7909	0.6163	**0.645**
IPA	0.8545	$0.676^{*}_{\_}$	$0.7383^{*}_{\_}$	0.5951	$0.5396^{***}_{\_}$
ASJP	0.8738	0.7241	0.7786	0.607	$0.5325^{***}_{\_}$
Metaphone	$\mathbf{0.9441}^{**}_{+}$	0.7265	**0.8005**	$0.5824^{**}_{\_}$	$0.4088^{***}_{\_}$
IPA 4-grams	0.9117	0.74	0.7895	$0.5611^{***}_{\_}$	$0.3793^{***}_{\_}$
P	0.8816	0.7178	0.7801	0.5979	$0.5655^{***}_{\_}$
PL	0.8664	0.7237	0.7734	$0.592^{**}_{\_}$	$0.5554^{***}_{\_}$
PLS	0.8006	0.7056	$0.737^{*}_{\_}$	**0.6165**	0.6197

this happens because the broader transcriptions reduce the information in an already small dataset even further. Subsequently, this circumstance weakens the validity of the results of this experiment overall. Still, except a surprising improvement from *Metaphone* by 7.31% in precision, all other all significant results are decreasing the performance of *verbatim* text. As the *c*@1-score of *Metaphone* is reduced by 23.62%, the most likely explanation for its raise in precision is that it produced many predictions within the uncertainty interval which were subsequently omitted in the determination of its precision.

Taking into account the results from all three experiments, we conclude that phonetically transcribing texts before using them in Authorship Verification does *not* increase the performance of Unmasking and the compression approach by any practically significant amount. Regarding the question whether a transcription systems granularity is correlated to its output, in the range of perfomance reduction, a slight trend can be seen: Methods that produce extreme amounts of tokens — many, such as *IPA* 4-*grams*, as well as few, such as *CV* 4-*grams* — perform worse than their more moderate counterparts.

In the following, we will discuss a number of possible reasons for the overall negative trend that phonetic transcriptions bring to the results. Converting graphemes to phonemes, in our case verbatim text to its IPA transcription, is a difficult task. Moreover, when transcribing automatically, the transcription algorithm does not have any information about the pronunciation of the speaker of a given text. Thus, usually text is transcribed to either the General (North) American pronunciation or the Received Pronunciation[4]. We use g2pE by Park and Kim [2019] which employs the Carnegie Mellon University Pronouncing Dictionary to look up transcriptions for words. The CMU dictionary uses North American English as its pronunciation standard. Thus, by transcribing we assume that the author has a North American English phonetic preference. Transcribing, for example, both an Irish author's and a Nigerian author's texts to American English, one can imagine that a lot of phonetically relevant information, that could be used to distinguish them, is lost. In the same vein, authors might *actively* make efforts to impart certain phonetic qualities into their texts that are based on topic rather than the author's unconscious phonetic preference. This becomes especially clear when an author uses direct speech, which often occurs in the datasets we use as they are both based on sets of fictional stories. The "voice in the author's head" when writing a direct speech passage presumably varies greatly depending on the traits of the character depicted in the story. Thus, extracting these features might aid

[4]British English

in topic or genre identification more so than in Authorship Verification. To mitigate this, direct speech could be removed from text entirely.

Another limitation of phonetic transcriptions for Authorship Analysis is due to the low-level nature they work at. An author's freedom of self-expression is limited on the sub-segmental level, i.e., concerning individual sounds. Usually, the meaning of a word changes together with its pronunciation. The only words for which this is not the case are synonyms such as the words "begin" and "start". Thus, authors that want to express similar ideas arguably will sound similar on the sub-segmental level not because of their phonetic preference but due to the proximity of the topics. More importantly, if authors want to express different ideas the resulting transcriptions will also be different, without the Authorship Verification classifier knowing if this difference stems from two unique authors with varied phonetic preferences or from one author discussing different topics. Supra-segmental features such as stress or prose can be utilized more freely and might give a more informative base for Authorship Analysis.

Lastly, standard Unmasking without using n-grams works on the lexical level, i.e., it uses entire words as atomic units and ignores the information encoded by the symbols inside a word. With the step of transcription, however, we are precisely attempting to enhance this inner-word information. Thus, for Unmasking the transcription of data serves only as a phonetically-informed binning method for types. In an effort to mitigate this, we used the n-gram-based transcription method, which still led to negative results.

Chapter 7

Conclusion

In this report we analyzed the viability of phonetically transcribing textual data prior to the use in Authorship Verification algorithms. Despite our initial expectations, we conclude that using phonetic transcriptions of data for Unmasking by Koppel and Schler [2004] and the compression approach by Teahan and Harper [2003] does *not* result in an improvement in performance. In fact, we observed many statistically significant decreases in overall performance. Nonetheless, we could identify a trend, namely the more a phonetic transcription system decreases the vocabulary size after transcription, the worse the performance of the algorithms. This may indicate that broader transcription systems lose more information that could have been useful for classification. The other extreme, inflating the vocabulary to 3.5 times its original size by generating 4-grams of the already detailed *IPA* transcription, also leads to a significant drop in performance across most of the measures analyzed. The only exception to the decrease in performance is the *Metaphone* algorithm in conjunction with the compression approach when using a sufficiently sized dataset, in our case one of the datasets introduced in Bevendorff et al. [2020a] comprising more than 50,000 training samples. For this configuration, we could record a slight improvement of 0.85% in precision over using verbatim text.

As is naturally the case with negative results, we cannot disprove that phonetically informed methods are generally not viable for Authorship Verification. We did, however, discuss some possible explanations for the absence of positive results from our research:

- During automatic transcription, too much phonetic information may be lost, as the transcription algorithm does not have any information about the specific pronunciation of the author.
- When working on the level of individual sounds, as we do with phonetic transcriptions, the sounds expressed when verbalizing an idea are often

tied to their meaning. Therefore, sub-segmental features might be better predictors for topic than for authorship.

- Unmasking works on the lexical level, because it treats words as atomic units and ignores the information encoded inside of them. This way, phonetic transcriptions are reduced to simple data reduction methods not exhausting their full potential.

Regarding possible improvements of our research, a more thorough inspection of the observed trend of diminishing performance with decreasing transcription granularity could lead to the substantiation of a correlation thereof. If other non-phonetically-informed token binning methods exhibit a similar correlation between performance and granularity, one could conclude that this trend is, in fact, not due to any phonetic phenomena. A different way of verifying this would be to down-sample the transcriptions so the vocabulary sizes are the same. Any remaining differences in performance cannot stem from the systems' granularities.

To achieve a better understanding of the utilization of phonetic transcriptions, verbatim and phonetically transcribed texts could be combined and feature selection algorithms could be used to identify the subset of features that is being employed by the algorithms in classification. This way, it could be determined whether AV classifiers utilize phonetically transcribed data over verbatim data.

Additionally to the investigations above, a detailed grid search for the best parameters for each transcription might lead to an improvement of the results.

For further research, we suggest a more profound investigation of the existence and nature of the *phonetic preference*. With a more thorough understanding of the ways people are influenced by their personal and subconscious phonetic palette, more meaningful feature extraction methods could be developed. These could include more in-depth approaches that, for example, disambiguate semantic information by analyzing certain punctuation structures within the text on the phonetic level. Also, shifting the focus from the low-level sub-segmental features to supra-segmental features spanning longer segments of text, such as stress, intonation, and rhythm, seems promising as authors have more freedom of self-expression on this level.

Bibliography

Anderson, C., Tresoldi, T., Chacon, T., Fehn, A.-M., Walworth, M., Forkel, R., and List, J.-M. (2018). A Cross-Linguistic Database of Phonetic Transcription Systems. In *Yearbook of the Poznan Linguistic Meeting*, volume 4, pages 21–53. Sciendo. 2.2, 4.2

Araujo-Pino, E., Gómez-Adorno, H., and Pineda, G. F. (2020). Siamese Network applied to Authorship Verification. In *CLEF (Working Notes)*. 3

Argamon, S. and Levitan, S. (2005). Measuring the Usefulness of Function Words for Authorship Attribution. In *Proceedings of the 2005 ACH/ALLC Conference*, pages 4–7. 3

Argamon, S., Whitelaw, C., Chase, P., Hota, S. R., Garg, N., and Levitan, S. (2007). Stylistic Text Classification Using Functional Lexical Features. *Journal of the American Society for Information Science and Technology*, 58(6):802–822. 3

Bevendorff, J., Ghanem, B., Giachanou, A., Kestemont, M., Manjavacas, E., Markov, I., Mayerl, M., Potthast, M., Rangel, F., Rosso, P., et al. (2020a). Overview of PAN 2020: Authorship Verification, Celebrity Profiling, Profiling Fake News Spreaders on Twitter, and Style Change Detection. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 372–383. Springer. 4.1, 7

Bevendorff, J., Ghanem, B., Giachanou, A., Kestemont, M., Manjavacas, E., Potthast, M., Rangel, F., Rosso, P., Specht, G., Stamatatos, E., Stein, B., Wiegmann, M., and Zangerle, E. (2020b). Shared tasks on authorship analysis at pan 2020. In Jose, J. M., Yilmaz, E., Magalhães, J., Castells, P., Ferro, N., Silva, M. J., and Martins, F., editors, *Advances in Information Retrieval*, pages 508–516, Cham. Springer International Publishing. 2.1

Bevendorff, J., Stein, B., Hagen, M., and Potthast, M. (2019a). Bias Analysis and Mitigation in the Evaluation of Authorship Verification. In *Proceedings of ACL 2019*. 2.1, 3, 4.1

Bevendorff, J., Stein, B., Hagen, M., and Potthast, M. (2019b). Generalizing unmasking for short texts. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 654–659, Minneapolis, Minnesota. Association for Computational Linguistics. 5.2, 6

Boenninghoff, B., Rupp, J., Nickel, R. M., and Kolossa, D. (2020). Deep Bayes Factor Scoring for Authorship Verification. *arXiv preprint arXiv:2008.10105*. 3

Brown, C. H., Holman, E. W., Wichmann, S., and Velupillai, V. (2008). Automated Classification of the World's Languages: A Description of the Method and Preliminary Results. *Language Typology and Universals*, 61(4):285–308. 2.2

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Lai, J. C., and Mercer, R. L. (1992). An Estimate of an Upper Bound for the Entropy of English. *Computational Linguistics*, 18(1):31–40. 5.1

Buringh, E. and Van Zanden, J. L. (2009). Charting the "Rise of the West": Manuscripts and Printed Books in Europe, A Long-Term Perspective from the Sixth through Eighteenth Centuries. *The Journal of Economic History*, pages 409–445. 2.1

Burrows, J. F. (1987). Word-Patterns and Story-Shapes: The Statistical Analysis of Narrative Style. *Literary & Linguistic Computing*, 2(2):61–70. 3

Cleary, J. and Witten, I. (1984). Data Compression Using Adaptive Coding and Partial String Matching. *IEEE transactions on Communications*, 32(4):396–402. 5.1

Dolgopolsky, A. B. (1986). A Probabilistic Hypothesis Concerning the Oldest Relationships Among the Language Families of Northern Eurasia. *Typology, relationship and time: a collection of papers on language change and relationship by soviet linguists*, pages 27–50. 2.2

Fossati, D. and Di Eugenio, B. (2008). I saw TREE trees in the park: How to Correct Real-Word Spelling Mistakes. In *LREC*, page 2008. 2.2

Hitt, J. (2012). Words on Trial. *The New Yorker*. 2.1

Howard, I. and James, P. (2019). Phonetic Spelling Algorithm Implementations for R. *Journal of Statistical Software, Forthcoming*. 2.2

IEEE (1969). IEEE Recommended Practice for Speech Quality Measurements. *IEEE Transactions on Audio and Electroacoustics*, 17(3):225–246. 2.2

International Phonetic Association, International Phonetic Association Staff, et al. (1999). *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press.

Kešelj, V., Peng, F., Cercone, N., and Thomas, C. (2003). N-Gram-Based Author Profiles for Authorship Attribution. In *Proceedings of the conference pacific association for computational linguistics, PACLING*, volume 3, pages 255–264. 3

Khomytska, I., Teslyuk, V., Kryvinska, N., and Beregovskyi, V. (2019). The Nonparametric Method for Differentiation of Phonostatistical Structures of Authorial Style. *Procedia Computer Science*, 160:38–45. 3

Koppel, M. and Schler, J. (2004). Authorship Verification as a One-Class Classification Problem. In *Proceedings of the twenty-first international conference on Machine learning*, page 62. 2.1, 3, 5.2, 7

Ladefoged, P. and Johnson, K. (2014). *A Course in Phonetics*. Cengage learning. 1

List, J.-M. (2010). SCA: Phonetic Alignment Based on Sound Classes. In *New directions in logic, language and computation*, pages 32–51. Springer. 2.2

List, J.-M. (2012). Multiple Sequence Alignment in Historical Linguistics. In *Proceedings of ConSOLE*, volume 19, pages 241–260. 2.2

List, J.-M., Greenhill, S., and Forkel, R. (2017). LingPy. *A Python library for quantitative tasks in historical linguistics*. 2.2

List, J.-M., Walworth, M., Greenhill, S. J., Tresoldi, T., and Forkel, R. (2018). Sequence Comparison in Computational Historical Linguistics. *Journal of Language Evolution*, 3(2):130–144. 4.2

Mendenhall, T. C. (1887). The Characteristic Curves of Composition. *Science*, 9(214):237–249. 3

Mosteller, F. and Wallace, D. (1964). Inference and Disputed Authorship: The Federalist. *Massachusetts: Addison-Wesley*. 3

O'Grady, W., Archibald, J., Aronoff, M., and Rees-Miller, J. (2017). *Contemporary Linguistics: An Introduction*. Bedford/St. Martin's, 7 edition. 2.2

Ordoñez, J., Soto, R. R., and Chen, B. Y. (2020). Will Longformers PAN Out for Authorship Verification. *Working Notes of CLEF*. 3

Park, K. and Kim, J. (2019). g2pE. `https://github.com/Kyubyong/g2p`. 4.2, 6

Peñas, A. and Rodrigo, A. (2011). A Simple Measure to Assess Non-Response. 6

Peng, F., Schuurmans, D., Keselj, V., and Wang, S. (2003). Language Independent Authorship Attribution with Character Level N-Grams. In *10th Conference of the European Chapter of the Association for Computational Linguistics*. 3

Philips, L. (1990). Hanging on the Metaphone. *Computer Language*, 7(12):39–43. 2.2

Potthast, M., Braun, S., Buz, T., Duffhauss, F., Friedrich, F., Gülzow, J., Köhler, J., Lötzsch, W., Müller, F., Müller, M., Paßmann, R., Reinke, B., Rettenmeier, L., Rometsch, T., Sommer, T., Träger, M., Wilhelm, S., Stein, B., Stamatatos, E., and Hagen, M. (2016). Who Wrote the Web? Revisiting Influential Author Identification Research Applicable to Information Retrieval. In Ferro, N., Crestani, F., Moens, M.-F., Mothe, J., Silvestri, F., Di Nunzio, G., Hauff, C., and Silvello, G., editors, *Advances in Information Retrieval. 38th European Conference on IR Research (ECIR 2016)*, volume 9626 of *Lecture Notes in Computer Science*, pages 393–407, Berlin Heidelberg New York. Springer. 5.1

Roser, M. and Ortiz-Ospina, E. (2016). Literacy. *Our World in Data*. https://ourworldindata.org/literacy. 2.1

Russell, R. C. (1918). Soundex. `https://worldwide.espacenet.com/patent/search/family/003328843/publication/US1261167A?q=pn%3DUS1261167`. 2.2

Russell, R. C. (1922). Soundex. `https://worldwide.espacenet.com/patent/search/family/024063815/publication/US1435663A?q=pn%3DUS1435663`. 2.2

Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to Information Retrieval*, volume 39. Cambridge University Press Cambridge. 4.3, 6

Smiley, C. and Kübler, S. (2017). Native language identification using phonetic algorithms. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 405–412. 3

Stamatatos, E. (2009). A Survey of Modern Authorship Attribution Methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556. 2.1, 3

Stamatatos, E. et al. (2006). Ensemble-Based Author Identification Using Character N-Grams. In *Proceedings of the 3rd International Workshop on Text-based Information Retrieval*, volume 36, pages 41–46. 3

Teahan, W. J. and Harper, D. J. (2003). Using Compression-Based Language Models for Text Categorization. In *Language Modeling for Information Retrieval*, pages 141–165. Springer. 3, 5.1, 7

Weerasinghe, J. and Greenstadt, R. (2020). Feature Vector Difference based Neural Network and Logistic Regression Models for Authorship Verification—Notebook for PAN at CLEF 2020. In Cappellato, L., Eickhoff, C., Ferro, N., and Névéol, A., editors, *CLEF 2020 Labs and Workshops, Notebook Papers*. CEUR-WS.org. 3

Williams, C. B. (1975). Mendenhall's Studies of Word-Length Distribution in the Works of Shakespeare and Bacon. *Biometrika*, 62(1):207–212. 3

Zobel, J. and Dart, P. (1995). Finding Approximate Matches in Large Lexicons. *Software: Practice and Experience*, 25(3):331–345. 2.2