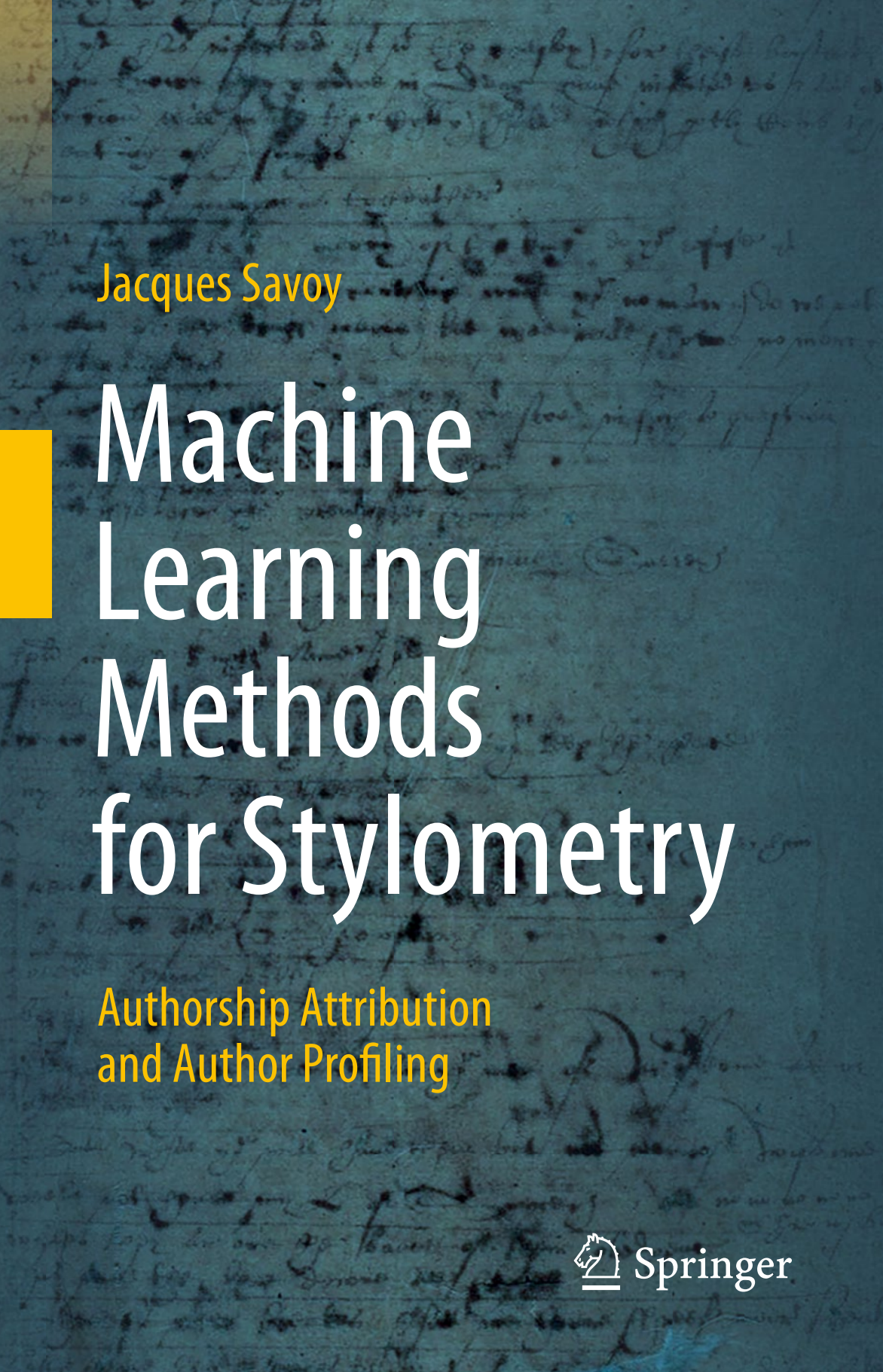
Jacques Savoy
Machine Learning Methods for Stylometry
Authorship Attribution and Author Profiling
Springer

# Machine Learning Methods for Stylometry

Jacques Savoy

# Machine Learning Methods for Stylometry

## Authorship Attribution and Author Profiling

Springer

Jacques Savoy
Department of Computer Science
University of Neuchatel
Neuchâtel, Switzerland

ISBN 978-3-030-53359-5 ISBN 978-3-030-53360-1 (eBook)
https://doi.org/10.1007/978-3-030-53360-1



This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*To Jacinthe, Adelaïde, and Benjamin*

# Preface

With the recent progress made in network and computing technology, the ubiquity of data, and textual repositories freely available, the scientific practice evolves towards a more data-based methodology. Thus, numerous domains consider machine learning models as pertinent tools to verify hypotheses or to improve their knowledge by discovering significant patterns hidden in datasets. And stylometry, or more generally digital humanities, follows this new research trend.

Focusing on the written style, this book presents methods and approaches able to identify the true author of a doubtful document or text excerpt. Assuming that each author has his[1] specific style, statistical or computer-based models can be applied to verify whether or not Shakespeare was the real author of a given play or poem. Besides literature works and authorship attribution, stylometric approaches can be useful to determine some demographics about the author. For example, one can wonder whether a novel (e.g., *My Brilliant Friend* (2012) by Elena Ferrante) is really written by a female writer. As other factors having a significant impact on the written style, one can study the effect of the author's age or his origin and native language. Instead of targeting the author, stylometric methods can be applied to draw the overall picture of style variations over a given time period or to underline the stylistic differences among a set of writers. With the ubiquity of social networks, stylometry can also be employed to infer some psychological traits of the author of a set of tweets as well as to identify early signs of depression. As a last example, stylometric measurements can be utilized to identify documents generated by machine or tweets sent by bots. This last aspect is related to the need to automatically detect fake news and its means and modes of dissemination.

Thus, the main intent of this book is to provide a broad introduction to all these text categorization problems grounded on stylistic features. This field of interest is clearly a multi-disciplinary one requiring some understanding in linguistics (or simply having some interest in this domain) and a basic knowledge in both statistics

[1] To simplify the presentation, the masculine form has been selected to indicate equivalently a man or a woman.

and computer science. We do not expect that the reader has an advanced skill in all these three domains. Thus, if the reader wishes to revise his knowledge in statistics, a gentle introduction, in plain English, is provided by Spiegelhalter [370]. As a solving-based approach is adopted in the different presentations of this book, explanations are supported by examples written in R. With S, its predecessor, this open-source software has radically changed the way statistics and data processing are applied; we moved from the pencil, calculator, and the use of various tables to statistical computing leading to modern data science. If the reader feels the need to acquire a better knowledge of R, an uncomplicated introduction is available in [41]. Other books expose the R software in a linguistics context (see [90, 182], or [139]).

## Book Structure

This book is subdivided into three parts. The normal reading sequence follows the chapter order. However, depending on the interest of the reader, some chapters could be skipped in a first reading. More precisely, the first part presents a general introduction and some well-known models for solving the authorship attribution question. This section is dedicated to readers having a background in the humanities. The second part (Chaps. 4–7) is more devoted to computer science with a focus on machine learning models. The third part corresponding to the last three chapters presents real stylometric applications and can be read by everybody. As sequential reading is not mandatory, some redundancy will appear from time to time.

In more detail, the first part covering the first three chapters proposes a general introduction to the stylometry domain with its possible applications and limits. After describing the main factors explaining written styles, our running example used to illustrate the presented concepts is exposed. Various overall stylistic measurements are defined and commented upon. Finally, Chap. 3 presents the four most frequently used stylometric modes to solve authorship attribution problems in the humanities.

Chapters 4–7 form the second part. As a fair evaluation methodology is crucial, this section starts with a chapter on this question. As this chapter contains more statistical arguments, it could be omitted in a first reading. As the main aim of this second part is to explain machine learning models for solving stylometric problems, a chapter exposes several general strategies to identify, extract, select, and represent stylistic markers. As fundamental models, this section presents the $k$-nearest neighbors ($k$-NN), the naïve Bayes, the support vector machine (SVM), and the logistic regression and applies them to our running example. The last chapter presents more recent approaches proposed to solve the authorship problem (e.g., the Zeta test, compression, latent Dirichlet allocation (LDA)). In addition, more specific methods have been developed for providing answers to more specific questions such as the verification issue (Is Shakespeare the author of the play *The Tempest*?) or to detect possible joint collaboration to write a novel. As the deep learning approach represents an active field of research, a presentation of neural network models and

word embeddings applied to stylometry is provided as well as a general introduction to the deep learning approach to solve stylometric questions.

The last part embraces the last three chapters, each of them focusing on a particular question. The main intent of this last section is to illustrate with real cases the application of the different approaches. When needed, complementary information can be obtained by following references to previous passages in this book. As application, Chap. 8 presents an authorship attribution problem, to know who is the secret hand behind the *nom de plume* Elena Ferrante, an Italian writer worldwide known for her *My Brilliant Friend*'s saga. The second real case concerns social media and more particularly the social medium platform Twitter. The subject is to verify whether a computer can identify if a set of tweets have been generated by a bot or a human being, and in this second alternative, if it was written by a man or a woman. The last application exposes various strategies to explore stylistic variations over time using US political speeches covering a period of around 230 years.

## Hands-On Exercises and Examples

To complement the presentation and discussion about stylometric models and techniques, examples and datasets are freely available. These illustrative examples are coded using the R software. This open-source language and the interpreter can be downloaded from the Internet at the following address:

`http://cran.r-project.org/`

It is important to know that the R software is used worldwide in statistics, both in academia and in industrial projects often related to big data applications. Knowing R is certainly a salient asset in your curriculum.

In addition, the datasets and the R code of our examples proposed in this book are freely available in the following GitHub webpage:

`https://github.com/JacquesSavoy/style`

For readers wishing to apply the presented methods on our examples or with other novels, we encourage the readers to download the `stylo` package written for R from the following URL:

`https://github.com/computationalstylistics/stylo`

This package provides useful stylometric functions and methods as well as additional novels to test them on. Our dedicated webpage also contains additional examples based on the `stylo` package to present advanced text representations or authorship models.

As a convention in this book, the `courier` font indicates statements, variables, or file names used in our examples written in R. *Italics* formatting is used when introducing important concepts but also to signal the novel or play titles and the occasional foreign terms. Finally, an index is available at the end of this book for a quick reference to the most important concepts.

Neuchâtel, Switzerland Jacques Savoy
May 2020

# Acknowledgements

Without discussions and research done by colleges, this book would not have been possible. First, I want to mention Prof. Dominique Labbé (University of Grenoble) for introducing me to the stylometry questions, methods, and evaluation measures. I also thank Prof. Arjuna Tuzzi (University of Padova) for organizing several summer schools on quantitative analysis of textual data IQLA-GIAT in Padova. She was also an important contributor inside the IQLA (International Quantitative Linguistics Association) and in developing the Elena Ferrante corpus (see Chap. 8). Of course, I should also mention other main contributors of the IQLA, namely, and in alphabetical order, Prof. Maciej Eder (Polish Academy of Sciences, Kraków), Prof. Patrick Juola (Duquesne University, Pittsburgh), Prof. George Mikros (National and Kapodistrian University of Athens), and Prof. Jan Rybicki (Jagiellonian University, Kraków).

My recognition also goes to the various people in charge for organizing all the CLEF PAN evaluation campaigns, to mention a few of them (in alphabetical order), Prof. Shlomo Argamon (Illinois Institute of Technology, Chicago), Prof. Moshe Koppel (Bar-Ilan University), Prof. Martin Potthast (Leipzig University), Dr. Francisco Rangel (Symanto Research), Prof. Paolo Rosso (Universidad Politecnica de Valencia), and Prof. Efstathios Stamatatos (University of Aegean).

# Contents

# Acronyms

Many acronyms and abbreviations are used in this book. For the most frequent ones, the following list provides the corresponding full name and for some of them a short definition.

| | |
|---|---|
| AA | Authorship attribution |
| ASCII | American Standard Code for Information Interchange |
| BW | Big word, word composed of six or more characters |
| CHI | Chi-square distribution |
| DNA | Deoxyribonucleic acid |
| FN | False negative |
| FP | False positive |
| FW | Functional words, corresponding to determiners, pronouns, conjunctions, prepositions, auxiliary and some modal verbal forms |
| GR | Gain ratio |
| HTML | HyperText Markup Language |
| IG | Information gain |
| LD | Lexical density, percentage of content-bearing words in a text |
| LDA | Latent Dirichlet allocation |
| LGBT | Lesbian, gay, bisexual, and transgender |
| LNRE | Large number of rare events |
| MeSH | Medical Subject Headings |
| MFL | Most frequent lemma |
| MFT | Most frequent word-type |
| MFW | Most frequent word, implicitly word-type |
| MSL | Mean sentence length |
| NN | Neural network |
| OR | Odds ratio |
| P | Precision |
| PMI | Pointwise mutual information |
| POS | Part-of-speech, or grammatical category or word class |
| QLF | Quadratic loss function |

| | |
|---|---|
| R | Recall |
| RNN | Recurrent neural network |
| RR | Reciprocal rank |
| SMS | Short message service |
| TC | Text categorization |
| TN | True negative |
| TP | True positive |
| TTR | Type–token ratio |
| URL | Uniform resource locator |
| WER | Word error rate |
| WWW | World Wide Web |
| XML | eXtensible Markup Language |

# List of Symbols

The following list indicates the main variables used in this book together with their definition. For example, the variable $n$ indicates the number of tokens occurring in a text, a sample of word-texts, or in a corpus (depending on the context). To represent the absolute occurrence frequency of the $i$th term, the variable $tf_i$ (where $tf$ means *term frequency*) is used. Depending on the context, this absolute frequency is computed only according to a single document or according to the entire corpus. The notation $tf_{i,j}$ is employed to indicate the absolute frequency of the $i$th term in the $j$th text. The variable $rtf_i$ denotes the *relative term frequency* of the $i$th term (with respect to a given document, author profile, or corpus).

| | |
|---|---|
| $\omega$ | An arbitrary word-type |
| $c$ | A constant |
| $Voc(T)$ | The set of word-types appearing in the text T (vocabulary) |
| $\|Voc(T)\|$ | The vocabulary size of text T |
| $Voc_k(T)$ | The set of word-types appearing exactly $k$ times in the text T |
| $\|Voc_k(T)\|$ | The number of word-types appearing exactly $k$ times in the text T |
| $n$ | The number of tokens (in the text) |
| $m$ | The number of stylistic features (terms) in a model |
| $r$ | The number of possible categories (or classes, authors) |
| ln() | The natural logarithm (and log() is logarithm with a basis = 10) |
| $func(T)$ | A function (to be specified) on the text $T$ (e.g., length($T$) = $n$) |
| $tf_{i,j}$ | The absolute term frequency of the $i$th term in the $j$th text |
| $tf_i$ | The absolute term frequency of the $i$th term |
| $rtf_{i,j}$ | The relative term frequency of the $i$th term in the $j$th text |
| $rtf_i$ | The relative term frequency of the $i$th term |
| $df_i$ | The absolute document frequency of the $i$th term |
| $r_i$ | The $i$th rank |
| $t_i$ | The $i$th term |
| $t_{i,j}$ | The $i$th term in the $j$th text |
| $D_j$ | The $j$th document |
| $p(t_i, D)$ | The occurrence probability of the $i$th term in the text D |

# Part I
# Fundamental Concepts and Models

The first part of this book covers the first three chapters and is intended for readers having a linguistic background, or more generally an interest in the humanities. Therefore, the statistical and mathematical notation is kept to the minimum needed, and the linguistic explanation or justification is viewed as more important. The main objective of the first part is to expose to the reader the diversity of problems that can be found under the general heading of stylometry. The observed written style of a document is not fully determined only by the author, but other important factors have key influences, for example, the text genre or the time period. For example, teenagers do not write a tweet like an essay, they do not speak between themselves as they speak with their grandparents. Those simple examples show us the large variability of communication situations and contexts even when analyzing the same author. Moreover, the author himself entails many dimensions that could be studied, such as the stylistic difference according to author's gender, age range, social origin, or psychological personalities.

Of course, the central application of stylometric studies remains the authorship attribution question, with the classic example being whether or not Shakespeare is the true author of his plays. But stylometric methods could also determine some demographic variables about the real author (e.g., Is it a woman? Older than 40 years old? Born in Germany?). The questions and their related concepts are presented and discussed in Chap. 1.

In Chap. 2, the notation and some additional concepts are introduced. Then a set of historical documents (the *Federalist Papers* written in 1787–1788) is presented and will serve as a running example for illustrating the different aspects of each stylometric model or measurement. Moreover, several overall stylometric measures are exposed and examined in order to be able to quantify the main aspects of a document, a given author, or a specific period. Numerical examples are introduced and commented to clearly understand the various steps appearing in the needed computation.

The last chapter of this first part presents in detail four authorship attribution methods usually applied in the humanities or with literary works. Each of these approaches precisely defines a stylistic representation of each text and suggests a

*modus operandi* to measure the stylistic similarity between pairs of texts or between author's profiles. Finally, they outline a decision rule or a procedure to identify the most probable writer behind a doubtful text or text excerpt. Of course, the presented methods could be applied to solve other questions than authorship attribution, for example, to draw a profile of the author or a stylistic map of a set of documents written in a given period by a group of writers.

# Chapter 1
# Introduction to Stylistic Models and Applications


During the past 50 years, the volume of data available in electronic format has grown exponentially with the progress accomplished in computer and network technology. In parallel, various statistical tools and methods have been designed, implemented, and are now freely available (e.g., with the R software and its numerous packages). Nowadays, numerous domains of human knowledge view digitized data as a valuable resource and apply machine learning approaches to verify hypotheses or to identify patterns in large datasets. Natural language processing (NLP) and some branches of applied linguistics are following the same direction. Thus the aim of this book is to precisely present and discuss different models and approaches for solving stylometric problems, in particular to solve authorship attribution questions, to discover the author's gender, or to resolve other stylometric questions.

All these questions can be solved by different text classification models, usually based on one sample of examples or instances for which the correct attribution or decision is known, and a second sample (sometimes limited to a single document) for which the attribution is either unknown or doubtful. The main problem is then to understand the most important factors explaining the style differences between authors or more generally between predefined categories such as author's gender, or time periods. Then one can define precisely how a computer system can represent the style of a text or a set of documents, how the similarity between two stylistic representations can be effectively measured, and what degree of certainty can be attached to the proposed attribution. This first chapter provides a broad overview of the different target applications and defines and explains the main concepts of stylometry.

The rest of this chapter is organized as follows. Section 1.1 defines the notion of text classification in the context of stylometric applications. Section 1.2 explains the concept of text style and describes its main explaining factors. Section 1.3 exposes the most well-known problem in stylometry, namely the authorship attribution question and three variants. The next section describes the author profiling question in which some demographics of the author should be inferred from the text



J. Savoy, *Machine Learning Methods for Stylometry*,
https://doi.org/10.1007/978-3-030-53360-1_1

s/he wrote. Section 1.5 illustrates some examples related to forensic linguistics. Section 1.6 exhibits the author clustering problem in which the system must recognize text belonging to the same class (e.g., written by men vs. women, or according to the true author). The last section reveals some additional problems or questions that can be solved, even partially, by considering the stylistic aspects present in a single text or in a sample of texts.

## 1.1 Overview and Definitions

The main objective of a *text categorization* (TC) or *text classification* system is to automatically assign predefined labels to texts according to their content or style. In this book, a *text* corresponds to a natural language writing not an audio source or a picture of a text (e.g., a scan of a medieval manuscript). The term text or document must be interpreted in a broad sense and can correspond to several forms (e.g., novel, poem, allocution transcript, last will letter, blog post, set of tweets, etc.). The text could be stored in a structured format (e.g., in XML) in which the document structure is clearly marked and the logical components (e.g., chapters, titles, footnotes) can be easily identified. With a semi-structured document (e.g., a web page with its HTML tags), the logical structure is partially provided while an unstructured text (e.g., a transcript of an uttered speech) can be viewed as a stream of words. The document structure could be useful for some applications and is not of prime importance for others. The target text could be limited to a part of an entire work (e.g., a chapter in a novel, a scene in a play, or even a few paragraphs in an e-mail). Associated with a document, one can find tables, graphics, pictures, videos, or hyperlinks. These non-textual elements could be useful in determining the true labels, but the current presentation is focusing mainly on the textual content.

Second, the *labels* represent the possible *categories* of interest. They must be viewed as tags without pertinent and useful meaning for the classification task. These labels could indicate the candidate author names (e.g., Shakespeare, Marlowe, Bacon), the possible text genres (e.g., play, poem, novel), the various keywords or topics, or even a binary answer (e.g., yes or no). When determining the possible labels, several scenarios can be encountered. The set of possible categories could be limited to two (e.g., Is this text written by a man or a woman? Is this document written by a single author?). Usually, the target labels form a set of possible answers and only one must be assigned to the test document (e.g., Is this text written in French, Italian, Spanish, or Portuguese? (language identification)). Sometimes, a few labels can be assigned to a document. For example, in a news agency, an incoming newsflash can receive several keywords such as "technology," "India," and "emerging market."

The set of predefined labels can form a more complex structure such as a tree (e.g., Is this article about sport, business, politics, or culture? And what kind of sport/business/politics/cultural event). In this context, usually more than one label, keyword, or descriptor can be attributed to the input text (e.g., exactly $k$ keywords, or up to a maximum of $k$). This last problem corresponds to the automatic indexing

process working with a controlled vocabulary (e.g., the Medical Subject Headings (MeSH) contains more than 25,000 descriptors structured into a thesaurus).

Text categorization applications can broadly be subdivided into two principal subdomains. First, the assignment can be performed according to the *semantics* of the document. The main objective is to help the user exploring a large volume of information (e.g., such as in the medical domain with PubMed via the MeSH thesaurus). Filtering is another pertinent tool in which a user can build his profile according to a set of keywords. Incoming documents (e.g., scientific articles, news, e-mails) are then analyzed by the filtering system, and when they correspond to a user profile, the selected texts are sent to the final user. Another classic example is spam filtering, removing non-relevant e-mails in mailboxes.

Second, the categorization process can be based mainly on the *text style*. The content itself is not of prime importance, but the text style forms the core from which pertinent features should be extracted. The style in its broad definition reflects personal choices, usually related to the main intent of the author (e.g., to explain and persuade the reader) but also for aesthetic reasons [395]. The particular style is reflected by the used words, the expressions, or the jargon occurring in the texts. By inspecting the sentence construction, the style includes aspects related to the syntax and grammar.[1] In addition, the repetition rate, the mean sentence length, and the frequent use of particular stylistic figures (e.g., ellipses, similes, metaphors, etc.) are other elements associated with the style. The author choices are not unlimited, and the adopted style is subject to constraints (e.g., oral vs. written communication, text genre, etc.). The style of a remark uttered by G. Washington is clearly distinct from Obama's style, and a colloquial conversation differs from a formal one or from the style that can be observed in a set of tweets.

## 1.2 Style and Its Explaining Factors

*Style* can be defined as a manner of expression or way of writing, starting with the choice of words, the combination of two or three words, the punctuation, the sentence structure, the target prosody, the grammar patterns, and all the elements that an author likes to use [301]. For Crystal [73]:

> "By style I mean the set of linguistic features that, taken together, uniquely identify a language user. The notion presupposes that there has been a choice—that someone has opted for Feature P rather than Feature Q (or R, S, ...)."

[1] The grammar specifies how the words are arranged to form sentences while the syntax governs the word order.

The linguistic items defining a particular style can be found at the lexical, syntactical, grammatical, and semantical level, as well as in the text layout. To determine a stylistic element, the notion of *choice* or *freedom* is essential. For example, synonyms offer multiple alternative words or expressions to indicate the same (or similar) concept (e.g., restaurant, coffee shop, bar, saloon, cafeteria, inn, pizzeria, Starbucks, where we met last week, etc.). Koppel et al. [217] suggest exploring this degree of liberty by evaluating a degree of "synonymy." The syntax also offers some freedom to the writer. Of course, for some items the position is fixed, for example, the position of the determiner *the* in the sentence "Now, the cat chases the mouse" (the sequence "mouse the" makes no sense). However, the position of the adverb *now* is not fully fixed and it can appear in another position ("the cat now chases the mouse" or "the cat chases the mouse now"). Furthermore, useful stylistic features are both *frequent* and *ubiquitous*. When writing a sentence and taking account of both the lexicon and the syntax, the author is faced with multiple decisions to achieve the wished target effect.

The stylistic markers do not appear only at the lexical and grammatical stage. At the semantics level, one can analyze the context of some words to define the particular idiosyncrasy of an author. For example, in Corneille's plays, the word *love* is strongly associated with the father's figure (who is usually an obstacle to this love) [228].

More concretely, can we differentiate between the style and the contents of a message? As a good example, Crystal [76] presents this set of sentences (see Table 1.1) reflecting several styles, progressing from a formal style to a casual one.[2]

**Table 1.1** Stylistic variations around the same content

| |
|---|
| The village does not have a post office. |
| The village has no post office. |
| The village doesn't have a post office. |
| The village hasn't got a post office. |
| The village hasn't got no post office. |
| The village ain't got no post office. |

The style and the contents must not be viewed as fully independent but as two faces of the same coin. The author selects a style to support a message and to reach an objective. However, in this choice several constraints must be taken into account.

As a first and most important explaining factor, the *text genre* explains some lexical or syntactical choices [34, 45]. Writing a sentimental or an adventure novel, an ode, a heroic couplet, a tragedy in verse, or a comedy in prose imposes some limits in the preference for some words or expressions. For example, the

[2] In French, Queneau [312] was able to write the same short story (around two paragraphs) in around 100 different ways.

composition of a poem is governed by a type of rhythm (and sometimes by a fixed number of syllables per verse) restricting the lexical choice of the author. When analyzing stylistic differences between text genres written by the same author, Burrows [45] concludes that different text genres written by the same author present more variability than texts belonging to the same text genre but written by several authors. This result was also confirmed by other studies [16, 30].

To illustrate this, we can mention scientists who prefer using the passive voice. Such constructions allow them to present the facts in a more impersonal and objective form (e.g., "it can be observed . . . ," "spectral analysis was applied . . . "). Comparing texts belonging to distinct text genres, similarities and differences can be discovered and the expression *style of poetry* or *newspaper style* reflects these stylistic resemblances within a given text genre.

The second factor is the *author* himself[3] with his own choice and background (e.g., gender, age range, education, social class, native language, etc.). Individuals have some likes and dislikes about the language and the writing. When facing with synonyms, some persons prefer one word or expression than another, for example, the term *actually* or *in fact* (other examples: while/whilst, because/since, film/movie, etc.). At the grammatical level, some authors produce longer sentences and frequently use the construction *of the*. Other writers prefer using contracted forms when possible. Some authors opt for longer explanations and describe all the details, while others choose concise descriptions. In applied linguistics, all these individual language differences are studied in *stylistics* while the variations between groups separated by social variables (e.g., gender, social position, region) are the object of *sociolinguistics* research.

The *time period* corresponds to the third factor [378] and each period imposes its own stylistic preferences. This aspect is visible in expressions like *classical style*, *postmodern style*, etc. For a more concrete example, one can observe that the sentence length decreases over time. In the eighteenth century, the mean length of speeches uttered by US presidents since 1945. For example, in speeches uttered by Madison under his presidency (1809–1817), the mean sentence length reaches 42 words per sentence while Trump's mean corresponds to 20.5 words [344]. This tendency seems to be reinforced by a fast-paced life (and for some persons by the frequent usage of texting or tweeting).

These first three factors correspond to the most important ones. As additional reasons, the topics have clearly some effect on the lexical choice and more generally on text style. It is known that we can encounter a medical, political, or legal parlance with each domain having its own vocabulary (or *lexis*), idiomatic expressions, and phraseology. When writing a novel, the author may have to provide the correct terms to describe a harbor or the words occurring in a dialogue between two sailors.

[3]To simplify the presentation, the masculine form has been selected to indicate equivalently a man or a woman.

The *communication type* also plays a role in the style. We do not write as we speak and when using web-based communication channels, we can adopt new features (e.g., less strict orthography, emojis) [74, 256]. In a tweet, one can write "C U" to say *see you* but the former will never appear in a newspaper article. In oral, the speaker pronounces more pronouns compared to a written text, repeats the same expressions more frequently, and thus presents a less abundant vocabulary [75].

Lastly, the *audience* has an influence on the style (e.g., look at the language differences between an official speech or a colloquial discussion) [187]. In an informal discussion, one can use *gonna* but will opt for *going to* in another context. One could also mention the editor as a possible source of stylistic variations, and its impact seems to be related to the text genre [326]. The punctuation can be the object of this variability because, for example, the usage of the comma is controlled by imprecise rules that can be fixed and imposed by the editor [78].

When taking into account all of these stylistic factors and their implications, we reach the conclusion that defining a Shakespeare's style reflecting his entire work is impossible. As for all authors (e.g., Goethe, Dante, Proust, etc.), writing a poem for a given audience, a play in prose or verse, a tragedy or a comedy, and in a specific time period and context imposes constraints on the word choice, syntax, or grammatical constructions. Thus, one cannot speak about a unique Shakespeare's style, but each author presents different stylistic facets.

Finally, three clarifications must be provided. First, the term *author* covers different aspects. In this book, the author is the person who composed the text with his corresponding lexical and syntactical choices. It is not the person who writes the selected words on a paper or using a word processor (the copyist or amanuensis). Likewise, the author is not the person who elaborates the scenario, the thoughts to be expressed (e.g., in a testimony), or the different characters or figures appearing in a novel or a play.

Second, the term *traditional authorship* must be interpreted as manual investigation to determine the true writer [246]. Our focus is on computational and statistical-based methods to solve various text categorization questions based on the text itself. In this case, the term *stylometry* covers this scientific activity. Unlike traditional authorship investigation, external proofs such as the historical context of the produced work, bibliographical evidence, or physical analyses (handwriting, watermarks, chemical analyses of the ink or paper) are ignored.

Third, even if our examples are given in English, the described methods and practices tend to work well for all natural languages, based on letters, syllabaries (e.g., Hangul (Korea), Katakana (Japan)), or sinograms (Chinese). To the best of our knowledge, we are not aware of studies demonstrating a real divergence in the effectiveness of methods resulting solely on the script difference between two languages.

## 1.3 Authorship Attribution

Authorship attribution or author identification [163, 183, 190, 219, 246, 278, 325, 373] is a well-known problem in stylistics and certainly the most studied question in stylometry. This problem can be stated as follows. Given a set of texts with known authorship, can we determine the author of a new unseen document? Under this general definition, three distinct contexts can be encountered.

First, the *closed-set* attribution problem assumes that the real author is one of the specified candidates from whom a sample of texts is available. This is the typical problem that can be found in research papers. Usually, the list of possible candidates has been determined from different external evidences (e.g., selecting because they are in the same text genre and presenting similar styles as the disputed document). In many cases, such a list contains a few names. For example, several studies suggest that behind Shakespeare's (1564–1616) signature one can discover another name. In this authorship debate, the most cited possible true authors are Sir F. Bacon (1561–1626) (the favorite candidate during the nineteenth century and the beginning of the twentieth century), E. de Vere (17th Earl of Oxford, 1550–1604), C. Marlowe (1564–1593), W. Staley (6th Earl of Derby, 1561–1642), or J. Florio (1553–1625) [104, 262, 388]. For others, one must not look for a single person behind Shakespeare's works but a group of authors. However, knowing that Shakespeare lived from 1564 to 1616, the candidacy of Marlowe or that of de Vere presents some temporal overlap issues. A similar question appears in French literature with the hypothesis that the most famous plays known to be written by Molière (1622–1673) were in fact written by P. Corneille (1606–1684) [227].

If these questions concern a set of works published in the Early Modern English era, other authorship issues focus on parts of a particular work. Knowing that several plays appear with two names (e.g., the play *The Two Noble Kinsmen* was written jointly by J. Fletcher and W. Shakespeare), the authorship question is to determine which scenes have been written by the first and the second author [65, 415]. As another example, it is admitted that the play *Henry VI* was, in part, a joint work between Shakespeare and Marlowe (mainly in Parts 1 and 2).

Second, in the *open-set* context, the real author could be one of the proposed authors or another unknown one. This case requires a more complex attribution method allowing a *don't know* answer due to lack of evidence, insufficiency, or failure of proof for all possible candidates. This problem instance is less studied in research papers, certainly because most of the attribution methods tend to provide an answer even when the stylistic evidence is rather weak and not fully convincing.

Third, the *verification* question provides a binary response as to whether a given author did in fact write a given text [214, 215, 218]. As input, the system has a sample of texts written by the unique candidate on the one hand, and on the other, the test document. This problem should be viewed as the more general one [221]. As soon as you can provide an automatic and effective model to solve this question, you can solve the two previous ones. In fact, having for each possible author a sample

of their writings, the two previous questions can be transformed into a sequence of verifications, one per candidate author.

One of the oldest authorship attribution questions is precisely a verification one: determining whether or not St Paul is the real author of the *Epistle to the Hebrews*. De Morgan [86] suggests using the distribution of word lengths (the number of letters per word) as a stylistic indicator to solve this problem. As a more recent problem, one can ask if McCartney is the single author of the song *In My Life* [131].

To solve these three distinct authorship questions, one important and often hidden hypothesis is the stability of each individual style denoted as stylistic fingerprint or stylistic idiosyncrasy. "Le style c'est l'homme" (the style is the man) said Comte de Buffon, suggesting both a stylistic stability and a way to discriminate between several authors. It is assumed that, for a mature person, his stylistic markers and language patterns will not undergo a large change during his life [224]. Such a *stable* stylistic identity must be interpreted as the style of an adult. For Joos [187], the language and style correspond to five periods in a human life, namely the baby language style, then the child, teenager, adult, and finally the elder one. Each of these frames presents its own language patterns and style (both in the lexical and syntactic constructions), but the mature phase is certainly the longest. Thus, it is a good practice not to compare texts written when the author was a teenager to those produced later. Moreover, when comparing two works, the time span must be taken into account when the publication date varies more than two decades. For example, Hoover [169] found that two novels written by the same author but with a difference in the publication date of more than 30 years are difficult to be assigned to the same writer (see also [50, 121, 171]).

As output, the system can provide a single name (or a binary answer in case of verification). However, it is more common to return a ranked list of names with a score indicating the belief or degree of certainty that the corresponding author is the true one. A clear and precise interpretation of such values is rather difficult for the user. Thus, within some methods, a probability can be estimated for each possible answer leading to a better information and clearer interpretation of the result. Finally, the proposed attribution should also be supported by a linguistic reasoning or by highlighting some language pattern similarities that cannot occur simply by chance.

## 1.4 Author Profiling

In some issues, the true name of the author is not of prime importance. The focus is on author profiling [317] or authorship characterization with the objective to identify some demographics about the writer such as his gender, age range, native language, social status, or even some of his psychological traits [21, 40, 280].

From those variables, the gender distinction might be viewed as the simplest one. The classification decision is binary and a relatively large amount of textual data can be collected. However, such a classification system can be effective only if

the writing style between genders does differ [97, 428] on the one hand, and on the other, if such differences can automatically be detected [297]. In addition, it must be recognized that there is a continuum in style between an extreme male and female figure. Moreover, it is not clear whether or not LGBT people can show other distinct writing variabilities compared to prototypical male and female figures.

Due to a large disparity during the language acquisition period, it is hard to have a good estimate of the age range for babies or children [299]. Therefore, the first age range considered by a profiling system is usually teenager (e.g., from 16 to 20 years old). As a simple binary discriminative model, two categories can thus be considered such as teenagers vs. older persons (e.g., 25 and more). In another perspective, four to five age ranges can be created (e.g., 18–24, 25–34, 35–49, 50–64, and 65+) assuming that stylistic markers can be detected to differentiate between these age ranges. Usually, however, the text contents reflect aspects that can be useful to discriminate between some age ranges (e.g., *marriage* is more related to the 25–34 age range, while *mortgage* or *children* could be associated with the 35–49 age range). For example, for writers born before 1950, the masculine form is employed to denote both genders while younger authors prefer to use more frequently the expression *he or she* or *s/he* (a third-person neutral pronoun does not exist in English). In addition, younger writers tend to depict a larger stylistic variability [291] and usually present more feminine stylistic patterns than older ones [299].

In Tables 1.2 and 1.3, two short blog posts illustrate the text style difference between a male and a female writer, as well as the author age range differentiation. Usually, readers do not have difficulty in identifying a teenaged woman as the author of the passage shown in Table 1.2. As explanation, one can argue the presence of emotions (e.g., ashamed, cry) for determining the author's gender. For defining the age range, a good and precise explanation is harder to put forward. One can argue about a jazz competition and that the words appear simple, belonging to a basic vocabulary, leading to a teenager behind the text depicted in Table 1.2.

**Table 1.2** A first example of a blog post. Written by a man or a woman and which age range?

| |
|---|
| Yesterday we had our second jazz competition. Thank God we weren't competing. We were sooo bad. Like, I was so ashamed, I didn't even want to talk to anyone after. I felt so rotten, and I wanted to cry, but... it's ok. |

With the example depicted in Table 1.3, the identification of the author's gender is more problematic but the age range seems easier (this is written by a male writer, between 25 and 35 years old). But knowing that pronouns are used more frequently by female writers and that nouns appear more often with males simplifies this identification task. Just counting the number of {*I*, *me*, *he*, *we*} on the one hand, and on the other the number of nouns (or the number of {*the*, *of*, *in*, *this*}), one can detect the author's gender. Of course, other features can discriminate between

**Table 1.3** Another example of a blog post. Written by a man or a woman and which age range?

| My gracious boss had agreed to let me have one week off of "work." He did finally give me my report back after eight freakin' days! Now I only have the rest of this week and then one full week after my vacation to finish this damned thing. |
|---|

a male and female such as a higher frequency of emotion words and negations for women, or swear expressions for men.

The time period has also a clear impact on the style. To illustrate this, read the text depicted in Table 1.4, a passage of a political declaration from the eighteenth century (written on July 4th, 1776, by T. Jefferson[4] ). Nobody speaks like this today. First, this excerpt corresponds to a single sentence, a very long one with 71 words. Second, many words are capitalized (e.g., "Course," "Nature," "Law"). Third, the tone is very formal and solemn providing reasons justifying an important decision.

**Table 1.4** Excerpt from the US declaration of independence

| When in the Course of human events, it becomes necessary for one people to dissolve the political bands which have connected them with another, and to assume among the powers of the earth, the separate and equal station to which the Laws of Nature and of Nature's God entitle them, a decent respect to the opinions of mankind requires that they should declare the causes which impel them to the separation. |
|---|

The language used can also reveal the geographical origin both at a national or international level. Obviously, the different accents play the most important role in identifying where a speaker comes from [79]. In a message, the vocabulary, the grammar, and the semantics might vary from one region to another, variations studied in dialectology. For example, the verb *to be*, the negation, and the different contraction forms show variability across England, from the *it's not* (in the North) to *ain't* (East Midland or South). The word order can also switch from *give it to me* (South-West) [71].[5] At the lexical level, the same object could have different denominations (e.g., "on vacation" in the United States, "on holidays" in UK, shop vs. store, post vs. mail, or line (railway) vs. track).[6] The spelling is also affected with the well-known differences between US and UK English (center vs. centre, color vs. colour, etc.) [84]. The grammar also diverges from one country to another

[4]T. Jefferson wrote the first draft of the independence declaration. After this document was edited by a committee of five members, and then by the whole Congress.

[5]All these language differences are a matter of critical judgments and could lead to a prestige scale and mockery (which have absolutely no linguistics support).

[6]Such differences appear in other languages such as in French with the term *mobile phone* called *cellulaire* in Montreal, *natel* in Geneva, or *portable* in Paris.

(e.g., with some Caribbean features as in "They going with two car," missing both the verb *to be* and the plural suffix). At the semantics level, the same word could cover different meanings, for example, the term *robot* meaning traffic light in South Africa. These language variations could differ between the official one and the usage. For example, even if Canada has adopted the UK standard English, in practice the US dialect is used.

Other aspects of the author could have an influence on the resulting style. The social position can be reflected by the choice of some words instead of others (*sick* vs. *ill*). Persons having a higher social status will employ a richer vocabulary and use more articles, prepositions, and longer words [75, 299]. The social group also has an influence on our language. In everyday language, one meets the expressions *journalist style* [19] (e.g., factual, and appealing to a wide audience) or *politician tone* [154, 155] (e.g., to achieve an angry/confident mood from the audience).

The native language of a bilingual writer can be discovered in a text. For example, a native French speaker tends to use words coming from a Latin stem more often instead of a Germanic synonym (e.g., adorable vs. lovely). In addition, this person would add a space before the question mark or the semi-colon punctuation symbol.

Finally, some psychological traits, ways of thinking, or the current mood of the author can be detected by inspecting his writings [280]. Sometimes one can call this the voice (or personality) of the author. For example, Pennebaker [299] indicates that angry people employ negative emotions more often as well as second-person (you) or third-person pronouns (he/she/it/they). They prefer the present tense. If sadness can be detected with these same words, sad persons tend to employ the past or future tense more than the present one. In a recent study, Noecker et al. [283] show that some psychological variables (judgment, feeling, extraversion) can be identified with a relatively high accuracy (around 85%) but others are clearly more difficult to predict (way of thinking, perception with an effectiveness around 60%). In a similar analysis, a person having depression or anorexia will depict some specific language patterns (e.g., a higher usage of *I/me/my*) and negative emotional words or anger expressions (e.g., *sad*, *cry*, *pain*), together with a decrease in the frequency of third-person pronouns [61, 143, 244, 245].

## 1.5 Forensic Issues

Identifying the author of a text can be useful to resolve offenses or to establish a criminal profile of a perpetrator (*forensic linguistics*). In this context, features present in the speech (voice printing analysis) or detected in a handwriting analysis can be pertinent to identifying the individual or to providing some demographics about the possible author. Even limited to writings, the cautious analysis of offender texts (e.g., ransom letter, threat e-mail) can reveal a few linguistic patterns to support police investigations [289, 290].

For example and as described in [291], the author of a threat message was identified by his systematic incorrect spelling of some terms or formulations (e.g., *alot*, *aswell* and the recurrent use of *stuff* instead of *staff*) and by writing "?!" instead of a single question mark. In another case, the occurrence of rare or very unusual words or expressions can establish evidence to discover the possible author (e.g., *covfefe*[7] by Trump (May, 2017) or *lodestar* used by an anonymous opponent layperson at the White House (Sept., 2018)).

In this kind of investigation, the main concern is the text length, usually short, for example, a few SMSs or tweets. In such circumstances, the application of identification methods is less predictive and the error rate could be too high to be admitted in a court of law [54, 64]. The stylometry analysis could however be useful for police officers, for example, by reducing the number of possible suspects (for an example, see [137]).

Other issues could be solved by considering the strong similarities between two texts or two passages such as plagiarism detection [10, 27, 379]. As a rule of thumb, to discover such an awful practice, one can consider that an identical sequence of five words between two passages is a strong indication that both are coming from the same source or than the second is a copy of the first one [291]. Even if the occurrence probability of an identical sequence of five words is not zero, it is very small, apart from some named entity denominations (e.g., former vice president of the United States of America, chief executive of the Royal Bank of Canada (in short, CEO of RBC)).

Other forms of plagiarism are more difficult to detect by a simple comparison. Instead of a simple copy/paste operation, the source could be text paraphrased or rewritten with a set of possible synonyms and changes in the sentence construction. For example, in his inaugural speech (1961), Kennedy said, "Ask not what your country can do for you, ask rather what you can do for your country," but a similar sentence was uttered by President Harding in 1923 "We need to be thinking not so much of what the country can do for us but what we can do for our country" [176].

An automatic detection tool needs therefore to operate with a soft matching algorithm. The aim is then to match similar sentence construction and meaning but written with some lexical variations. In addition, the source message could be written in one language and the plagiarism in another, usually by simple machine translation [307]. With numerous text repositories, search engines, and freely available machine translation tools, the Web has facilitated all these activities. As a corollary, the demand for automatic detection systems of such practices has also increased.

[7] According to *The Independent*, *covfefe* means *coverage* and the full tweet was "Despite the constant negative press covfefe." The White House Press Secretary Sean Spicer confirms it was not a typo, but "the president and a small group of people know exactly what he meant" [110].

## 1.6 Author Clustering

Stylometry can be applied to solve both the *author clustering* and *authorship linking* problem. The latter is defined as follows. Having a sample of documents (or text excerpts) written by an unknown number of authors, determine the pairs of documents written by the same person. When an author wrote just one text, this one must stay unpaired.

With the author clustering question, the objective is to determine the number $k$ of distinct authors behind the sample of $n$ texts. In addition, the system must form $k$ distinct clusters, each grouping all texts written by the same writer.

As possible applications for both problems, one must pair or cluster a set of poems, political speeches, proclamations produced by several terrorist groups, or a collection of customer reviews written by different authors [9]. Of course, instead of identifying the author, the same scenario can be applied by clustering texts according to author demographics, text genres, or written during the same time period.

As an example of author clustering, Fig. 1.1 displays four clusters of documents written by four distinct authors. On the left, the attribution system detects all authorship links between the four texts in this cluster. This is a perfect answer. For the second cluster in the middle, the system discovers only some links shown by solid lines, for example, between Text 2G and 2Y. By transitivity however, one can deduce that all texts in this cluster are written by the same person. For example, knowing that Text 1W was written by the same author as Text 2G and that Text 2G and 2Y were composed by the same writer, one can deduce that Text 1W has the same author as Text 2Y.

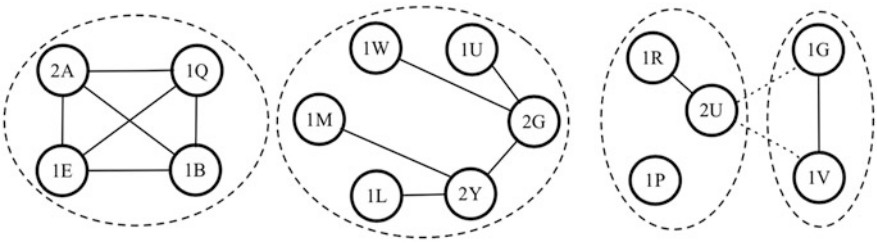


**Fig. 1.1** Author clustering with four clusters for four authors

For the last two clusters on the right, the system incorrectly specifies that Text 2U and 1G or 2U and 1V have been written by the same author. In Fig. 1.1, these two incorrect links are indicated with a dotted line. Finally, Text 1P was left as an orphan and the system assumed that the corresponding author just wrote this one. In fact, this writer also wrote Text 1R and 2U while the fourth creator only wrote Text 1G and 1V.

To solve these tasks, it is not possible to learn the different styles in advance to choose the most appropriate subset of stylistic features. The author names (or other target categories) are unknown as well as their number. The system does not have any learning dataset in the form of a sample of texts written by different authors. The entire information is provided by the sample of texts to be clustered. Therefore, numerous machine learning models cannot be applied.

To design such a system, an *unsupervised* approach must be adopted (without any training data). The focus is to determine an effective intertextual distance (or similarity) measure based on a predefined set of stylistic characteristics. Such a measure can then be applied between pairs of texts (authorship linking) or groups of texts (author clustering).

## 1.7 Other Related Problems

Related to the classical authorship question, one can ask if a given literature novel is the work of a single author or the fruit of a collaborative effort. This last question could be completed by providing the number of writers (and their names), as well as the specification of the text passages written by each author. As an example, one can mention the heroic epic poem *Beowulf* written in Old English (between 975 and 1025[8] ). As for many works from this era, it is an anonymous writing. For scholars, the question is to know whether this poem was written by a single author or not. After analyzing it, J.R.R. Tolkien [396] specifies that this work corresponds to a unique author. A recent study [279] tends to confirm this hypothesis of a unique writer.

As another issue in stylometry studies, one can mention the detection of forgeries or fabricated stories. When writing, does a liar produce some distinctive linguistic patterns, using some words or expressions more frequently and ignoring others? According to Vrij [418] and Pennebaker [299], the answer seems positive but a reliable detection is not so simple. Language is flexible and many formulations can produce the same or a similar meaning (without the same style and tone).

One of the oldest discoveries of a fake document is related to the *Donation of Constantine*, a manuscript transferring broad powers to the pope over Rome and Western Europe. This decree of the emperor Constantine the Great (272–337) was viewed as essential by the Catholic church to justify the control by the pope over Rome and Italy. In 1439–1440, the humanist Lorenzo Vella analyzed the language present in this manuscript and discovered the forgery. Based mainly on lexical elements, Vella picked out some anachronistic expressions (e.g., usage of terms appearing only in the eighth century) or unusual usage of some pronouns. The essential characteristic of this finding is the analysis of some linguistics elements

[8]Other experts argue that this manuscript could be older and date from the eighth century.

together with their frequencies instead of arguing with external evidences and authoritative sources.

As more recent examples, one can cite reporters able to write faked stories. For example, a journalist (C. Relotius) from the German newspapers *Der Speigel*, winner of numerous awards, invented chronicles or protagonists in at least 14 over 60 of his articles [59]. The *New Republic Magazine* faced a similar problem with one of their columnists (S. Glass) who wrote 27 articles containing faked stories or fabricated material. But can an automatic system detect a lie?

A liar has a tendency to produce short sentences, using short words. His story exposes less details, less numbers, and includes more certainty expressions or tone [299, 418]. Thus, expressions such as *except*, *I believe*, *I think* are usually missing or occur with a low frequency. The automatic detection of a written lie is still hard, and the best success rate is around 60%. Other examples related to social networks and micro-blogging can be found in [359].

Many other problems can be analyzed using stylometry models and large datasets. With the current technology, accessing a huge number of documents is no longer a real challenge . For example, Google *Ngram Viewer* [261] allows us to access around 4% of all printed books from 1800 by sending a query (a word or a list of words). With such a tool, one can observe the relative frequencies of various terms across a time interval, for example, the increasing use of some technologies (e.g., phone, computer) or the decrease of others (e.g., steam engine, telegraph). Linguistics can see the variation in usage of some synonyms such as radio and wireless or some syntactical change. For example, during the first part of the nineteenth century, the noun *United States* implies a verb in the third person plural while in the beginning of the twentieth century, the verb appears more often in singular ("The Unites States are" vs. "The Unites States is"). Using this website, Juola [192] demonstrates that one can measure quantitatively the increasing complexity of the Western culture. In another study, the same author demonstrates how one can quantify the linguistic evolution of language [189].

As another example, a set of experiments on literary history is described by Jockers [180]. In this case, the author presents a set of available techniques to analyze the corpora of literary works, for example, the word clouds or the Latent Dirichlet Allocation (LDA) techniques (see Sect. 7.3) for content analysis, the distribution of genre over time, and the distinction between genders according to the vocabulary used (see also [175]).

Analyzing the individual language differences with quantitative tools leads to other applications related to political science. For example, one can compare the rhetoric and the style of all US presidents from Kennedy to Obama [155] or from Washington to Trump [344] (see Chap. 10), or to contrast the various styles adopted by the candidates during the primaries election [345] or during the general election by comparing D. Trump's vs. H. Clinton's style [348]. Such questions are presented with more details in Chap. 10.

# Chapter 2
# Basic Lexical Concepts and Measurements


To solve the various authorship attribution problems described in the first chapter, the proposed approaches must be grounded on quantitative linguistics and statistical models using computers to store the texts and to perform the needed computations. This paradigm requires that the style of a given author must be represented by one or a sequence of numbers (usually stored in a vector). The stylistic features of a text are present at various levels such as in the vocabulary, the syntax, the grammar, the semantics, and in some cases in the layout, presentation, etc. Usually, it is not possible to consider all those aspects together. Moreover, one must assume, implicitly or explicitly, that some linguistics items should be considered as more important than others to portray an author's style. Those selected style markers should therefore be detected inside the document and transformed into numbers.

Another important point must be underlined. As mentioned previously with Vella's discovery, stylistic identification is usually performed on a *comparative basis*. The attribution model compares the style representation of the disputed text with those of the candidates (or other categories such as male or female, teenager or adult, or corresponding to several time periods, etc.).

The aim of this chapter is to introduce different basic concepts and it is organized as follows. The first section introduces the notion of stylistic models. Section 2.2 exposes our running example called the *Federalist Papers* exploited in our presentation to explain, describe, and evaluate different stylometric models and approaches. Section 2.3 presents the Zipf's Law and its implications in analyzing the most frequent and rare words used by an author. Section 2.4 exposes various vocabulary richness measures, while the next section describes some overall stylistic measurements. The last section briefly exposes methods based on letter occurrence frequencies to characterize different authors' styles.



J. Savoy, *Machine Learning Methods for Stylometry*,
https://doi.org/10.1007/978-3-030-53360-1_2

## 2.1 Stylometric Model

To define a *stylometric model*, three steps must be clearly explained. First, the style of the disputed document and those of text samples written by known authors (or corresponding to the different categories) should be extracted and represented in an efficient and effective manner. This surrogate generation must be clearly described without ambiguities. The difficult aspect is to ensure that all essential stylistic markers are taken into account in the chosen style representation. During this first stage, the target system usually represents each text by one surrogate corresponding to one point or one vector (*instance-based* representation). However, all texts corresponding to the same category (e.g., written by the same person, or reflecting the same gender, age range, etc.) can be regrouped to generate a single author or unique category *profile*.

Second, a measure must be chosen to compute the distance (or similarity) between the representations of the test text and the different authors or categories (instance-based) or the distinct author or category profiles (profile-based). Usually, the result of this computation is a single number reflecting the closeness between two text representations or profiles. Therefore, the selected measure combines or aggregates the different stylistic markers into a single value.

Third, a ranked list of possible candidates or categories must be returned to the user. This list could be limited to a single name and even include the answer *don't know* when the computed assignment does not reach a given threshold of certainty. Of course, the value of the distance or a degree of belief can be provided with each proposed name or category label. When the interpretation of such a value is rather clear, this information should help the user to interpret the proposed attribution.

Before describing different models and approaches, a precise definition of the term *word* must be provided. For example, how many words do you count in the sentence "I saw a man with a saw."? Various possible answers can be provided. The first answer counts the number of *word-tokens* (or simply tokens) that refer to an occurrence or instance of a word. In this case, the answer is seven, or even eight when considering the punctuation as a token. The second answer counts the number of distinct words or *word-types*. This corresponds to the vocabulary present in the sentence. In this case, the answer is five because one can see two occurrences of *a* and two of *saw*. As the third answer, one can consider that the two instances of *saw* do not correspond to the same *lemma* (headword or entry in the dictionary). Therefore, this sentence contains six lemmas (I, (to) see, a, man, with, saw), without taking account of the punctuation symbol.

An important final comment. A term is not the synonym of a word. A term corresponds to a stylistic feature such as a word-type (e.g., we), a specific Part-Of-Speech (POS) tag (e.g., a personal pronoun), a sequence of two words (bigrams of words) (e.g., "we can") or, in general, an $n$-gram of words (a sequence of $n$ consecutive words, such as "it is in our"), as well as $n$-grams of POS tags (e.g., adj-noun-noun). In other models, the stylistic characteristics could be based on single characters (e.g., "o") or on sequences of $n$ characters ($n$-grams of letters or

characters) (e.g., "th") as well as the presence or absence of some logical part (e.g., the commercial letter text genre is characterized by the presence of an address and greetings).

## 2.2 Our Running Example: The *Federalist Papers*

To illustrate the concepts and methods presented in this book, a running example is always useful. In this aim, the *Federalist Papers* [203] represents an interesting corpus. It is composed of 85 newspaper articles from which 12 are disputed between two possible authors (Hamilton[1] and Madison[2] ). These texts were written to persuade "the People of the State of New York" to ratify the Constitution [248] and published (and republished) between October 1787 and May 1788 in newspapers under the pseudonym of *Publius*. Under this name, Alexander Hamilton (1755–1804), James Madison (1751–1836), and John Jay (1745–1829) have jointed their efforts to present the merits of the new Constitution and to answer critics formulated by the anti-federalists [206].

If, at the time of publication, the authorship of each paper was kept secret, contemporaries have guessed the joint work of Hamilton, Madison, and Jay, without being able to explicitly attribute each article to its legitimate author. In 1804, two days before his fatal duel, Hamilton gave the first assignment (called the Benson's list, see Table 2.1). In this list, there is a large consensus agreeing that a substitution occurs between the author's name of Paper #54 and #64. We will adopt this position and admit that Hamilton wrote Paper #54 (instead of Jay as specified in the Benson's list) while Jay is the author of Paper #64 (instead of Hamilton). After his presidency in 1818, Madison gave his assignment, revealing 15 differences between the two lists. The last column of Table 2.1 indicates the current admitted attribution of each article. This position reflects a large consensus, but some authors do not share this attribution, for example, Rudman [327] who suggests that the disputed papers are jointly written by Hamilton and Madison.

In Table 2.1, 70 articles are undisputed (5 by Jay, 14 by Madison, and 51 by Hamilton) and they form the training set from which the distinct stylistic features of each writer can be defined. Three papers have been written jointly by Hamilton and Madison (Paper #18 to #20) and they will be ignored. The limited contribution of Jay could be explained by his illness during the winter 1787–1788 and thus, for some analyses, Jay's involvement could be ignored. In the disputed or test set, 12 papers appeared that could have been written by either Hamilton or Madison (Paper #49 to #58 and #62 to #63).

This collection owns pertinent characteristics for an authorship test corpus. First, all these articles are from the same text genre (newspapers articles) with the same

[1] Alexander Hamilton was the first Secretary of the US Treasury (1789–1795).

[2] James Madison was the fourth US president (1809–1817).

**Table 2.1** Authorship of the 85 *Federal Papers* according to Benson's list, Madison, and the current attribution

| Number | Benson | Madison | Current |
|---|---|---|---|
| 1 | Hamilton | Hamilton | Hamilton |
| 2–5 | Jay | Jay | Jay |
| 6–9 | Hamilton | Hamilton | Hamilton |
| 10 | Madison | Madison | Madison |
| 11–13 | Hamilton | Hamilton | Hamilton |
| 14 | Madison | Madison | Madison |
| 15–17 | Hamilton | Hamilton | Hamilton |
| 18–20 | Madison & Hamilton | Madison | Madison & Hamilton |
| 21–36 | Hamilton | Hamilton | Hamilton |
| 37–48 | Madison | Madison | Madison |
| 49–53 | Hamilton | Madison | Madison |
| 54 | Hamilton (Jay) | Madison | Madison |
| 55–58 | Hamilton | Madison | Madison |
| 59–61 | Hamilton | Hamilton | Hamilton |
| 62–63 | Hamilton | Madison | Madison |
| 64 | Jay (Hamilton) | Jay | Jay |
| 65–85 | Hamilton | Hamilton | Hamilton |

objectives. Second, they were published during the same time period (around a one-year period). We can also assume that the same spelling and punctuation have been imposed for this set of articles. Fourth, the main topics were very similar across all articles, specifically an explanation of the new US constitution. The main factor varying from one article to another is the author. Moreover, this corpus was investigated by Mosteller and Wallace [273] whose contribution is viewed as the first modern application of statistically based methods in authorship attribution.

Finally, written by three Founding Fathers, this corpus is also interesting for historical reasons. These commentary papers on the principles of government are still an essential source of interpretation for the US Constitution [203, 260]. For example, in the eleven essays written by A. Hamilton about the powers of the presidency, some passages have been used to clarify the power of impeachment in 2019–2020 [8].

**Table 2.2** Statistics about the three authors of the *Federalist Papers* (training set)

| Author | Number | Size (tokens) | Vocabulary |
|---|---|---|---|
| Hamilton | 51 | 110,924 | 6919 |
| Madison | 14 | 38,765 | 4234 |
| Jay | 5 | 8383 | 1725 |

Table 2.2 depicts some overall statistics[3] about the 70 articles appearing in the training set, namely the number of articles per author, their length in number of tokens, and the vocabulary size (number of distinct word-types). To illustrate the complexity of a correct assignment of the 12 disputed documents, one can compare the occurrence frequencies of some Part-Of-Speech (POS) tags. On the one hand, when comparing the percentage of determiners used in Hamilton's vs. Madison's articles, the difference is rather small (17.8% by Hamilton vs. 17.9% by Madison, a difference of 0.1%). On the other hand, when comparing Hamilton's with Jay's papers, the difference is larger and rises to 5.8% (17.8% vs. 11.9% for Jay). For nouns, a similar pattern can be detected. The difference in percentage between Hamilton (22.8%) and Madison (22.2%) is small (0.6%), while with Jay (20.6%) the difference is larger (2.2%). The largest difference between Hamilton and Madison appears with the word *to* used to indicate the infinitive (Hamilton: 4.4%, Madison: 3.5%, difference: 0.9%; Jay: 3.7%). Clearly Hamilton's and Madison's styles are closely related, while Jay's style is more distant, at least according to the percentages of some POS tags.

## 2.3 The Zipf's Law

For Biber and Conrad [31], a stylistic study should be based on ubiquitous and frequent forms. But how can the computer extract and represent the style of a given author, text genre, time period, or general category? As an operational procedure, the aspects reflecting a style could be defined as the words used by that category (e.g., specific author, text genre, period, author gender, etc.) together with their frequencies. However, such an operational definition limits the style to its lexical component. Such a solution could be accepted, at least as a first approximation.

Using this viewpoint, can the stylistic differences be detected by comparing the most frequent word-types? As an example, the 10 most frequent word-types (excluding punctuations symbols)[4] used by the three authors of the *Federalist Papers* are shown in Table 2.3. For each word-type, the absolute frequency (denoted $tf_{i,j}$) is provided together with their relative frequency shown in percentage. As reported in Table 2.3, the definite determiner *the* is the most frequent in this corpus, with a relative frequency between 10.0% (Madison) and 6.2% (Jay). The presence of the prepositions *of* and *to* is also a characteristic or a trademark of the Modern English language. A closer look at word-types present in Table 2.3 indicates that they contain between two and four letters, illustrating the principle of least effort in human communication. Frequent forms are short and thus fast to pronounce or to write.

[3] Small differences occur with the values reported in Table 2.2 when punctuation symbols and numbers are also counted as tokens.

[4] In the Appendix, one can find Table A.2 with the same information computed with the punctuation symbols.

**Table 2.3** Ten most frequent word-types per author (*Federalist Papers*)

| | Hamilton | | | Madison | | | Jay | | |
|---|---|---|---|---|---|---|---|---|---|
| Rank | Type | $tf_{i,j}$ | % | Type | $tf_{i,j}$ | % | Type | $tf_{i,j}$ | % |
| 1 | the | 10,186 | 9.2 | the | 3876 | 10.0 | the | 516 | 6.2 |
| 2 | of | 7106 | 6.4 | of | 2306 | 5.9 | and | 408 | 4.9 |
| 3 | to | 4478 | 4.0 | to | 1247 | 3.2 | of | 359 | 4.3 |
| 4 | in | 2773 | 2.5 | and | 1163 | 3.0 | to | 288 | 3.4 |
| 5 | and | 2671 | 2.4 | in | 808 | 2.1 | in | 164 | 2.0 |
| 6 | a | 2472 | 2.2 | a | 768 | 2.0 | be | 160 | 1.9 |
| 7 | be | 2270 | 2.0 | be | 754 | 1.9 | that | 150 | 1.8 |
| 8 | that | 1679 | 1.5 | that | 542 | 1.4 | it | 138 | 1.6 |
| 9 | it | 1523 | 1.4 | it | 497 | 1.3 | as | 102 | 1.2 |
| 10 | is | 1296 | 1.2 | is | 481 | 1.2 | a | 100 | 1.2 |

This table indicates that the same 10 most frequent word-types appear under the pen of Hamilton and Madison. The ranking is identical except one swap between the 4th and 5th rank. In contrast, Jay's ranking presents some differences and the word *as* occurs in the top 10 (but not *is*). Overall, the same word-types appear as the most frequent for these three authors. This finding confirms the previous POS distribution analysis indicating a very close relationship between Hamilton's and Madison's style.

Another interesting aspect of this list is the text coverage of those 10 most frequent types. For Madison, for example, the word-types *the* and *of* represent 15.9% of all tokens. Together, the top 10 most frequent word-types cover 32.9% of all articles written by Hamilton, or 32.1% for Madison, and 28.5% for Jay.

When analyzing the relation between the rank (from the most frequent to the less frequent) and the frequency (absolute or relative), Zipf [432] found that the frequency observed at the $i$th rank (denoted $tf_{r,i}$) is inversely proportional of the rank ($r_i$) at the power $\alpha$ [14]. (But Zipf estimates that $\alpha$ must be equal to 1.) This relationship is described in Eq. 2.1 in which $c$ is a constant:

$$tf_{r,i} = \frac{c}{r_i^{\alpha}} \text{ or } tf_{r,i} \cdot r_i^{\alpha} = c \tag{2.1}$$

Taking the log of the two parts of Eq. 2.1, we obtain a linear relationship between the log of the term frequency and the log of the rank as depicted in Eq. 2.2. In this formulation, the value of the parameter $\alpha$ corresponds to the slope of the relation shown in Fig. 2.1.

$$\log(tf_{r,i}) = c - \alpha \cdot \log(r_i) \tag{2.2}$$

Based on Hamilton's writings, Fig. 2.1 displays this Zipfian relation (with c = 9.6 and $\alpha = -1.03$). In addition, the least squares estimation is depicted with a dashed red line. This estimation is close to perfect in the middle of the values but

tends to overestimate the frequencies in the two tails. Thus, the label *law* could be viewed as too strong, and some authors prefer speaking about Zipf's regularity. Of course, more complex Zipfian models have been proposed to allow a better fit with the frequency distribution [14, 341].

Instead of looking at the most frequent word-types (MFWs), one can analyze the number of word-types occurring just once in the corpus (denoted *hapax legomena*), or twice (*dis legomena*) as reported in Table 2.4. This table indicates that 38.6% of all entries in Hamilton's vocabulary appear only once (*lexis* size: 6919) while 14.7% appear twice. In addition, 5036 of the word-types in the vocabulary occur five times or less in Hamilton's papers.

**Table 2.4** Frequency of word-type according to their absolute frequency

| Frequency | Hamilton | Madison | Jay |
|---|---|---|---|
| $tf_{i,j} = 1$ | 2672 (38.6%) | 1932 (45.6%) | 972 (56.3%) |
| $tf_{i,j} = 2$ | 1017 (14.7%) | 752 (17.8%) | 284 (16.5%) |
| $tf_{i,j} = 3$ | 625 (9.0%) | 379 (9.0%) | 133 (7.7%) |
| $tf_{i,j} \leq 5$ | 5036 (72.8%) | 3459 (81.7%) | 1515 (87.8%) |

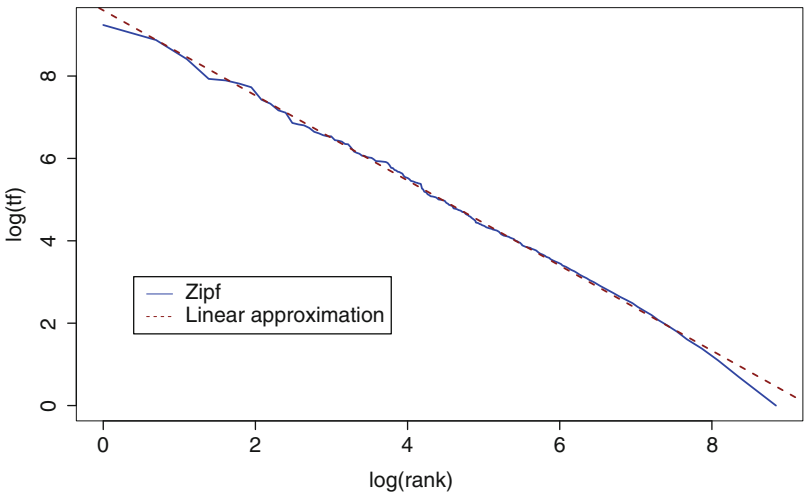


**Fig. 2.1** Absolute frequency as a function of the rank (log scale, Hamilton's articles)

Analyzing the relationship between the frequencies of tokens (see Table 2.3) and the distributions of the word-types (see Table 2.4), one can find two interesting findings. First, a few word-types occur very often (see Table 2.3) and cover a large part of all tokens present in a corpus (e.g., the top 10 most frequent covers 32.9% of Hamilton's writings). Those words will be associated with stylistic markers in

many stylometry models because they are both ubiquitous and very frequent in all texts (and all languages). More important, their occurrence frequencies vary from one author or category to the other.

Second, to achieve the full coverage of a corpus, one needs to consider a large number of word-types occurring rarely. According to values shown in Table 2.4, the percentage of word-types appearing once or twice is larger than 50% of the total. This phenomenon is called the *Large Number of Rare Events* (LNRE) [14]. All languages present such a distribution characterized by a large number of very low probability elements. Therefore, it is impossible to define all words belonging to a language or an author. As soon as one considers an additional passage, new unseen word-types will appear for the first time. Those new terms could be spelling errors, new names (e.g., geographical, personal, products), or new words (e.g., to google, chatbot, post-truth, alt-right, Brexiteer), as well as new associations between words (e.g., carbon footprint, glass cliff, low-carbon electricity).

## 2.4 Vocabulary Richness Measures

Various vocabulary richness or lexical diversity measures have been proposed [15, 163] to indicate the amplitude of the vocabulary (or *lexis*) used by an author or a speaker. The most well-known measure is the TTR (Type-Token Ratio) or the ratio between the number of word-types (vocabulary size) and the number of tokens (text length) [15, 270, 303]. The definition is indicated in Eq. 2.3 where the text length (number of tokens) is denoted by the variable $n$ and the set of word-types appearing in the text by Voc(T):

$$TTR(T) = \frac{\mid Voc(T) \mid}{n} \tag{2.3}$$

High values indicate the presence of a rich vocabulary showing that the underlying text exposes many different topics or that the author writes on a few themes from several angles with different expressions and formulations. On the other hand, a small TTR value signifies that the vocabulary used by the author is limited or that the ideas expressed in the text are repeated with the same or very similar expressions. This ratio reflects some stylistic aspects of the target text and in particular, the vocabulary richness or lexical diversity.

It has been proposed to use the TTR values as an authorship attribution method, assuming that each author could have his own distinct TTR value. With the *Federalist Papers* corpus, and based on data shown in Table 2.2, Hamilton presents a TTR of 0.062 (= 6919/110,924), while Madison has a TTR value of 0.109. The last writer, Jay, has the highest TTR value at 0.206. If a new text presents a value close to 0.1, the system will assume that the true author must be Madison.

But such an assignment strategy presents some drawbacks. First, the TTR value is sensitive to the text length. As this length increases, the resulting TTR

decreases [14]. By observing how the vocabulary grows along with a text, one can observe that the author has a tendency to reuse previous terms and the rate of occurrence of new words tends to decrease without reaching zero [341].

Second, when extracting a sequence of chucks of the same size (e.g., 5000 words) from a text, the resulting TTR values are not exactly the same but will depict some variability. Thus, an author's style cannot be described by a single number but by a range of possible values.

To illustrate these notions and findings, Fig. 2.2 indicates the (global) TTR for Hamilton (red dashed horizontal line) at the bottom of the figure (value 0.062). In the middle, one can find the TTR obtained by non-overlapping chunks of 5000 tokens (a blue decreasing line). To draw this line, the first point indicates the TTR obtained with the first 5000 words, the second point with the first 10,000 words, the third with the first 15,000 words, etc.

On the top part is the evolution of the TTR computed after each segment of 5000 tokens (green dashed line on the upper part). In this case, the first point indicates the TTR computed with the first 5000 words, the second point with words occurring between 5001th position and 10,000, the third estimates with tokens in position 10,001 to 15,000, etc. For this last line, a confidence range is also depicted by two dotted line ($\pm$ 2 $\times$ the standard deviation).

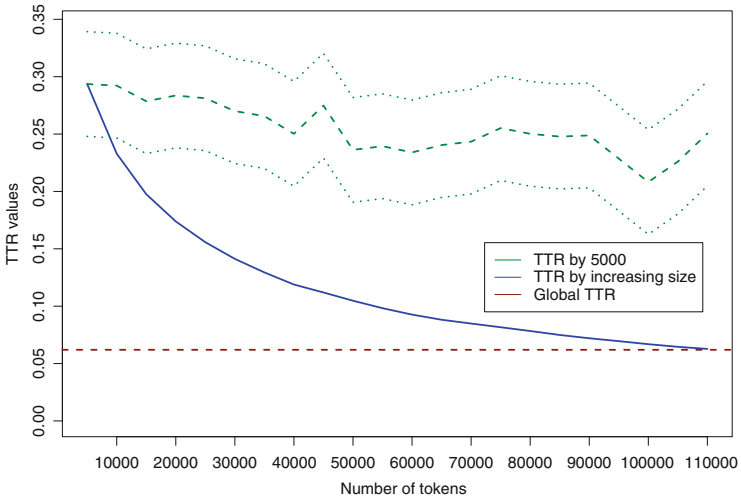


**Fig. 2.2** Evolution of the TTR values (Hamilton's articles)

Figure 2.2 shows clearly that the TTR values decrease when the text length increases (see blue line "TTR by increasing size"). When considering the same text length (5000 in Fig. 2.2), the resulting TTR values ("TTR by 5000") expose some underlying variability [409]. To partially avoid the instability of the TTR measurement, a better computation is provided by Popescu et al. [303] who suggest

taking the moving average of the TTR. With this additional correction, the TTR value can reflect some aspects related to the vocabulary richness but cannot be an effective strategy for authorship attribution.

TTR is not the unique measure suggested to quantify the lexical diversity nor has it been suggested as a single value able to characterize the style of an author. Guiraud's R [141, 142] and Herdan's C [157] are also well-known measurements of the vocabulary complexity (see Eqs. 2.4 and 2.5). In the first case, the denominator corresponds to the square root of the text length while in the second equation the natural logarithm ($ln()$) is applied to both the vocabulary size and text length.

$$Guiraud's\ R(T) = \frac{\mid Voc(T) \mid}{\sqrt{n}} \tag{2.4}$$

$$Herdan's\ C(T) = \frac{\ln \mid Voc(T) \mid}{\ln n} \tag{2.5}$$

The density of *hapax legomena* (word-types appearing once) or the *dis legomena* (word-types occurring exactly twice) has also been used to measure some of the lexical diversity of a given text or author. These two measures are depicted in Eq. 2.6 for the *hapax* density while the Sichel's S [361] is shown by Eq. 2.7. Taking the average over multiple non-overlapping chunks of the same size, the *hapax* density combined with other measures of lexical diversity was applied to approximatively date ancient writings [367].

$$Hapax\ density(T) = \frac{\mid Voc_1(T) \mid}{\mid Voc(T) \mid} \tag{2.6}$$

$$Sichel's\ S(T) = \frac{\mid Voc_2(T) \mid}{\mid Voc(T) \mid} \tag{2.7}$$

Found to be the more stable measure of vocabulary richness (e.g., independent of the text length) [14, 409], Yule's K (1944) [266, 387] and Simpson's D (1949) [362] are however slightly more complex to compute (see Eqs. 2.8 and 2.9). The main idea behind both formulations is the following. When a word-type $\omega$ appears $r$ times in a corpus of size $n$, the probability to randomly select one instance $\omega$ is $\frac{r}{n}$. The probability of selecting twice the same word-type $\omega$ is $\frac{r}{n} \cdot \frac{r}{n}$ (with replacement, see Eq. 2.8) or $\frac{r}{n} \cdot \frac{r-1}{n-1}$ (without replacement, see Eq. 2.9). In addition, in a corpus, the number of word-types occurring $r$ times is given by $|Voc_r(T)|$. Both indicators are based on the repeat rate; the higher this value, the lower the lexical richness.

$$Yule's\ K(T) = c \cdot \left[ -\frac{1}{n} + \sum_{r=1} \frac{r}{n} \cdot \frac{r}{n} \cdot \mid Voc_r(T) \mid \right] \quad \text{with} \quad c = 10^4 \tag{2.8}$$

$$Simpson's\ D(T) = \sum_{r=1} \frac{r}{n} \cdot \frac{r-1}{n-1} \cdot \mid Voc_r(T) \mid \tag{2.9}$$

For having a better understanding, let us take an example. Having a corpus containing $n$ times the same word-type (thus $r = n$), the Simpson's D value is equal to $\frac{n}{n} \cdot \frac{n-1}{n-1} \cdot 1 = 1$, which is the maximum value, indicating a very low vocabulary richness. However, when all word-types appear exactly once, the intern summation of the Yule's K is $\frac{1}{n} \cdot \frac{1}{n} \cdot n = \frac{1}{n}$. From this, the fraction $\frac{1}{n}$ must be subtracted, producing 0. Therefore for both indicators, a small value denotes a rich vocabulary.

Table 2.5 depicts these vocabulary complexity measures for the three authors of the *Federalist Papers* by averaging over a sequence of non-overlapping segments each containing 5000 tokens. As usually the highest values indicate the largest vocabulary,[5] Hamilton appears to be this author with Guiraud's R and the *hapax* density indicator. According to Sichel's S or the mean word length, Madison is the writer having the more complex lexicon while the TTR, Herdan's C, Yule's K, or Simpson's D favor Jay.

**Table 2.5** Vocabulary richness measure of the authors of the *Federalist Papers*

| Measure | Hamilton | Madison | Jay |
|---|---|---|---|
| TTR | 0.2545 | 0.2478 | 0.2664 |
| Guiraud's R | 17.994 | 17.522 | 17.087 |
| Herdan's C | 0.8389 | 0.8358 | 0.8410 |
| Hapax density | 0.6271 | 0.6099 | 0.6084 |
| Sichel's S | 0.1566 | 0.1664 | 0.1620 |
| Yule's K | 184.02 | 188.08 | 126.34 |
| Simpson's D | 0.0184 | 0.0188 | 0.0126 |
| Mean word length | 4.792 | 4.878 | 4.848 |

None of these measures has proven very satisfactory as an authorship attribution scheme [167, 409], due in part to word distributions (including word bigrams or trigrams) ruled by a large number of very low probability elements (*Large Number of Rare Events* or LNRE) [14]. When analyzing word frequency distributions, one can see that they diverge from more familiar distributions (e.g., Gaussian, Poisson, or binomial distribution). As soon as more text is taken into account, new unseen words appear leading to the conclusion that it is impossible to generate a complete set of all words belonging to a given language or author. This concern could be marginal if they cover only a few percentages of the vocabulary. However, the percentage of such rare words occurring once or twice represents around 50% of the entire vocabulary (see Table 2.4).

[5] Be careful, the interpretation is the reverse for both the Yule's K and Simpson's D.

## 2.5 Overall Stylistic Measures

As stylometry covers multiple applications, several measures have been proposed without having a direct implication to an authorship attribution approach. The lexical density (LD) [32, 286] is a good example. With a value varying from 0 to 100%, this measure indicates the percentage of lexical items (or content-bearing words) appearing in a text (or in a dialogue). A higher value signifies a more informative text.

As shown in Eq. 2.10, the lexical density is the ratio between the number of lexical items (value returns by the function $content(T)$) and the text length indicated by $n$.

$$LD(T) = \frac{content(T)}{n} = 1 - \frac{functional\ words(T)}{n} \tag{2.10}$$

As content-bearing words, linguists count the number of nouns, verbs, adjectives, and adverbs. The remaining POS categories correspond to functional words and include the determiners (e.g., the, a, those), the prepositions (e.g., of, in), the conjunctions (e.g., and, if, but), the pronouns (e.g., we, him, mine), and some auxiliary verb forms (e.g., is, been, had, must, can, will). These kinds of words do not have clear lexical properties, and possess more grammatical-syntactic functions. One can view them as the needed glue between lexical items. As another definition, the functional words contain closed POS categories while the lexical items enclose open POS classes or categories for which new words can be generated. If one can discover new nouns, verbs, adjectives, or even adverbs, the apparition of a new pronoun,[6] determiner, or preposition is rare[7] [256]. Thus Eq. 2.10 can be written as shown in the right part by counting the number of functional words in a given text. To be complete, some linguists opt for removing the adverbs from the content-bearing items, and thus the resulting LD values could be smaller in that case.

A careful look at a text reveals other difficulties. If the punctuation and other symbols (e.g., $, €) are ignored (and also do not count in estimating $n$), names and foreign words can count as nouns as well as abbreviations (e.g., UNO). The question is less clear with some homographs having two or more distinct POS categories and meanings. The word *to* is a preposition, but it is also present in front of a verb in infinitive form (e.g., to see). The same concern appears with verbs having a particle such as to *give up*. Do we ignore the *to* and *up* or count them as a functional word? Is the genitive suffix "s" (e.g., in Paul's book) count as a functional word? Different authors might propose different answers to these questions.

When using the lexical density, some comparative values must be provided. According to [291] and [185], one can ascertain, in average, an LD value to be

[6] For example, it would be useful to have a gender-neutral pronoun instead of the s/he and her/him.

[7] For example, the pronoun *thou* (2nd person, singular) disappears in the Modern English and has been replaced by the *you*.

around 0.3 as a norm for an oral production and 0.4 and higher for writings (or even 0.5 or more for a scientific message). Based on the *State of the Union* speeches (a formal remark of the US president in front of the Congress), [344] found LD values between 0.4 and 0.5, the highest value obtained under Eisenhower's presidency (1953–1961). The lexical density can also be a useful measurement for other classification tasks. It tends to increase with the author's age, and it is usually lower for oral production than for writings [138, 185].

The style complexity can also be measured by the percentage of *big words* (BW) (e.g., words composed of six letters or more) [389]. A text or a dialogue with a high percentage of big words tends to be more complex to understand. This fact is confirmed by recent studies in neuroscience:

> "One finding of cognitive science is that words have the most powerful effect on our minds when they are simple. The technical term is basic level. Basic-level words tend to be short. ... Basic-level words are easily remembered; those messages will be best recalled that use basic-level language." [232, p.41].

Producing a text with a high percentage of BW implies that the underlying style is sophisticated, looks scientific, and is more complex to comprehend. This is usually the opposite of the intent of a politician during an election campaign. For example, during the 2016 US presidential primaries campaign, Savoy [345] indicates that the percentage of big words varies from 18.3% (Trump) to 26.4% (Cruz, or Sanders). The winner opted for a simple and direct speech, one easy to follow.

An older method is related to the proportion of BW: it is the mean word length and the word length distribution. Augustus de Morgan [86] suggests applying this measurement as an authorship attribution method. Due to the lack of computing resource, the application was not possible at that time. Later Mendenhall [258] proposed to analyze the frequencies of different word lengths when inspecting Shakespeare's plays. According to his computation, he suggests that Marlowe could be the secret hand behind Shakespeare's works because for both authors, the most frequent word length is four (e.g., compared to three for Bacon).

As another stylistic indicator, the mean sentence length (MSL) reflects some syntactical choice done by the author [251]. The sentence boundaries are defined by strong punctuation symbols (namely periods, question marks, and exclamation marks). Usually, a longer sentence is more complex to understand, especially in the oral communication form. Analyzing the *State of the Union* addresses, the Founding Fathers presented a high value, in mean, 39.6 tokens/sentence [344]. Under Obama's presidency, the MSL decreased to 18.5 tokens/sentence. These examples clearly indicate that the style is changing over time. Nowadays, the preference goes to a shorter formulation, simpler to understand for the audience. During the 2016 US primaries election, the MSL confirms this tendency with an average close to 19

for all candidates, except for Trump who is adopting an even more simple and direct communication style (MSL = 13.7). The presence of long sentences (e.g., H. Clinton, B. Sanders) indicates a substantiated reasoning or specifies the presence of detailed explanations. Even if a long sentence is required, its length does not guarantee an easy understanding.

With the ubiquity of web-based communication channels, other overall stylistic markers have been proposed such as the percentages of uppercase letters (particularly useful with Tweets), the percentage of emojis and emoticons, hashtags, retweets, or hypertext links (more on this topic in Chap. 9).

## 2.6 And the Letters?

During the Italian Renaissance, Leon Batista Alberti (1404–1472), a well-known architect but also the first western cryptograph, was the first to suggest using a quantitative measurement in stylometry. Alberti's problem was to distinguish between prose and poetry in Latin. In the latter, the proportion of vowels is larger than 7/16 (or 43.75%). For the orators, when dividing the number of consonants by the number of vowels, one can find a proportion smaller than 4/3. Applying this strategy to a corpus of work written in Latin, Ycart [426] confirms the usefulness of Alberti's remark. If Alberti can be viewed as the first person using a quantitative measurement with style, a more detailed presentation of the history of stylometry can be found in [163, 246].

Another example of the use of letter frequencies can be found in [237]. The objective is to identify the true author of each of the 26 scenes present in the play *The Two Noble Kinsmen*, a work written jointly by Shakespeare and Fletcher. As spelling in the Elizabethan era is not fully stable, the pair of letters "i" and "j" is considered as the same, as well as the letters "u" and "v." In addition, the letters "q," "x," and "z" have been ignored due to their low frequencies, reaching zero for some text samples. The stylistic differences correspond to frequencies of the remaining 21 letters in text samples of 25,000 letters.

In a related study, Merriam [259] demonstrated that the frequency of the letter "o" can discriminate between works written by Shakespeare and those of Marlowe. In fact, Marlowe's plays tend to have less than 7.8% of "o," while Shakespeare employed this letter more frequently. The linguistic reasoning behind such stylistic markers is rather simple. A more detailed presentation of this topic can be found in Sect. 5.2.

# Chapter 3
# Distance-Based Approaches


This chapter exposes a set of methods developed mainly inside the stylometry community with strong relationships to the humanities and literature studies. The main objective of these approaches presented in this chapter is to identify the true author of a disputed text. Instead of applying these models solely as authorship attribution methods, one can view them as a practical solution to author profiling problems (determining the author's gender or age range), as well as other text classification tasks (identifying the text genre or date of publication). Methods more related to computer science and machine learning are exposed in Chap. 6.

As explained previously, each stylometric model specifies a procedure to identify the main stylistic characteristics and to represent them for both the sample of texts reflecting the different authors and the disputed document. Finally, a distance (or similarity) measure is computed between the distinct style surrogates and the query text. Based on these measurements, the stylometric model indicates how one can derive an authorship attribution decision.

The rest of this chapter is organized as follows. Section 3.1 exposes the well-known Delta model suggested by Burrows [46] and serves to illustrate the term selection strategy, the weighting of these selected features, and the intertextual distance computation. Some possible variants of this model are also introduced. Section 3.2 describes a related model based on a fixed set of features and estimating the intertextual distance by comparing two probabilistic distributions based on the Kullback–Leibler divergence (KLD). In Sect. 3.3, Labbé's model is introduced, depicting another way for selecting terms and computing a distance. Section 3.4 presents other distance or similarity measures that can be applied with the previous models. Finally, the Principal Component Analysis (PCA) is briefly presented, mainly in order to project different styles info a 2D map.



J. Savoy, *Machine Learning Methods for Stylometry*,
https://doi.org/10.1007/978-3-030-53360-1_3

## 3.1 Burrows' Delta

To determine the real author of a text, Burrows [46] suggests accounting for the most frequent word-types (MFWs) without taking punctuation marks or numbers into consideration. In his original method called Delta, Burrows proposed considering from 40 to 150 most frequently occurring word-types, with 150 words usually obtaining the best effectiveness.

With this approach, the style is mainly reflected through the word choice, and more precisely by those very frequent ones in the entire corpus. This solution presents interesting properties and advantages. First, as shown with Zipf's law (see Sect. 2.3), these very frequent items mainly correspond to function words, used mainly unconsciously by the author and thus suitable to reflect his style. Second, those word-types do not have a precise meaning leading to a stylistic representation independent of the topics of the underlying texts. Third, these frequent lexical items can cover from 50 to 65% of all tokens occurring in a document or in a sample of documents. With the *Federalist Papers*, the 150 MFWs cover 64% of Hamilton's articles, or 65.2% for Madison. Thus, even limited to 150 items, the text coverage is rather large.

The style representation in the Delta model matches well with Biber and Conrad's definition [31] indicating that a stylistic study should be based on *ubiquitous* and *frequent* forms. Burrows places an emphasis on the lexical level and chooses to base the style representation on words. Such a choice is not really new, and other studies have proposed to take account of frequently occurring words as style indicators, the first one dating from 1975 [83, 267, 331].

But words might be ambiguous. Instead of counting similar spellings as the same term, Burrows suggests to distinguish between *homographs*, for example, between *that* as a conjunction or as a relative pronoun or *to* as a preposition or used with a verb in the infinitive base form. Adding these POS tags for each token requires an additional preprocessing. It is not sure that an automatic solution (e.g., based on a POS tagger [351, 398]) can provide a high-quality solution without some manual corrections.

When ignoring the ambiguity of the homographs and the corresponding POS tagging, this selection criterion is rather simple to apply, and the computational cost is low. In a follow-up study, Savoy [339] found that a term selection strategy based on the occurrence frequency is an efficient one compared to other possible selection methods (see Sect. 5.3). As indicated by Burrows [46], the proposed threshold of 150 MFWs is somewhat arbitrary and other values can be specified. Hoover [168] suggests to increase this threshold up to 800 words or 1000 in [372]. In another study, the range of effective limits goes from 100 to 500 MFWs, with, usually, the best performing values between 200 and 300 MFWs [339]. In determining a good threshold, the sample of texts must be taken into account. For example, Burrows [46] is working with 25 poets of the English Restoration. This corpus is not really large with around 540,000 words (and with each text having more than 1500 words), a size similar to the novel *War and Peace* by Tolstoy or twice the size of *Moby Dick*

by Melville. Savoy's study utilizes larger corpora (around 3,500,000 words), and the proposed limit is higher (e.g., 300) than the 150 in Burrows' case.

Could the punctuation symbols be stylistic features? If the punctuation rules are clear to denote the sentence end, the comma offers more freedom to the author. Some writers tend to put commas wherever possible (e.g., "Fortunately, the bus was on time, so Sheema wasn't late for the concert."), others opt to ignore them as much as possible (e.g., "Fortunately the bus was on time so Sheema wasn't late for the concert") (examples from [76]).

When considering other languages [331], previous studies tend to indicate that the most effective thresholds are similar to those found for English. One can argue that the definite article in English is the single word *the* and its translation in other languages possesses numerous forms (e.g., *le*, *la*, *les*, *l* in French, or *der*, *die*, *das*, *den*, *dem*, *des* in German). The same phenomenon appears with auxiliary verbs having more distinct forms in other languages than in English. However, if a language offers more diverse verbal forms that does not mean that all those forms are frequently used. Different combinations of tense, person, and number can generate more distinct verbal words in French than in English, but many of them are simply rare. Therefore, they might not be so useful to discriminate between distinct authors or stylistic categories.

Within the Delta model, the style associated with each text is represented by the frequency of the $m$ MFWs, defined by considering the whole text collection. However, each text does not have the same length. It is not appropriate to directly compare the absolute frequencies between texts not having a similar length. Therefore, instead of using the absolute term frequency (denoted $tf$), one can employ the relative term frequency ($rtf$). In fact, Burrows [46] goes a step further. Instead of using the relative term frequencies, he proposes to compute their standardized scores.

These values (denoted Z score) are obtained by subtracting the mean and then dividing by the standard deviation [168]. This score for each term $t_i$ (word-type) in a text sample (corpus) is computed by first computing the term's relative frequency $rtf_{i,j}$ in a document $D_j$ (or a text written by author $A_j$). Then one can subtract the mean ($mean_i$) and divide by the standard deviation ($sd_i$) for this term $t_i$ according to all texts belonging to the underlying corpus (see Eq. 3.1).

$$Z\ score(t_{i,j}) = \frac{rtf_{i,j} - mean_i}{sd_i} \tag{3.1}$$

To illustrate this computation, four articles written by Hamilton (prefixed by the letter H) and four by Madison (denoted by M) have been extracted from the *Federalist Papers*. To simplify the presentation, not all 150 MFWs have been taken into account but only eight (leading to a toy-size example). The punctuation symbols have been taken into account because we think they can be useful to reflect the author's style. For example, a writer with shorter sentences will present a larger number of periods (or full stops).

The absolute term frequency of each term according to the eight articles are reported in Table 3.1. For example, one can count 175 occurrences of the definite determiner *the* in Text H59 written by Hamilton. As another view, the column under the label "H59" is a vector representation of the stylistic aspects of this article. The last row of Table 3.1 shows the sum of these eight word-types and approximates the article size.

**Table 3.1** Absolute frequencies of the eight frequently used words (Hamilton and Madison)

| Word | H59 | H60 | H61 | H65 | M37 | M38 | M47 | M48 |
|---|---|---|---|---|---|---|---|---|
| the | 175 | 221 | 149 | 218 | 228 | 269 | 325 | 167 |
| , | 133 | 152 | 104 | 134 | 192 | 233 | 219 | 157 |
| of | 110 | 143 | 98 | 129 | 157 | 188 | 185 | 98 |
| to | 72 | 86 | 60 | 84 | 84 | 116 | 64 | 53 |
| . | 45 | 47 | 32 | 47 | 74 | 94 | 86 | 56 |
| in | 62 | 79 | 47 | 51 | 62 | 62 | 62 | 46 |
| and | 34 | 36 | 25 | 37 | 101 | 94 | 86 | 51 |
| a | 49 | 53 | 35 | 44 | 56 | 92 | 35 | 35 |
| *sum* | 680 | 817 | 550 | 744 | 954 | 1148 | 1062 | 663 |

In Table 3.2, using the same lexical items, the relative term frequency ($rtf$) is computed for each feature and article. For Text H59, the conjunction *and* has a value of 0.05 (= 34/680). In the last two columns of Table 3.2, the mean and standard deviation ($sd$) have been computed for each word-type. These values are computed according to the relative frequencies obtained by the eight texts. For the term *and*, this mean is 0.067. The results shown in Table 3.2 indicate that Hamilton tends to use this conjunction less than Madison (e.g., its relative frequency is 0.044 in H60).

**Table 3.2** Relative frequencies of the eight selected terms (Hamilton and Madison)

| Word | H59 | H60 | H61 | H65 | M37 | M38 | M47 | M48 | *mean* | *sd* |
|---|---|---|---|---|---|---|---|---|---|---|
| the | 0.257 | 0.271 | 0.271 | 0.293 | 0.239 | 0.234 | 0.306 | 0.252 | 0.265 | 0.025 |
| , | 0.196 | 0.186 | 0.189 | 0.180 | 0.201 | 0.203 | 0.206 | 0.237 | 0.200 | 0.017 |
| of | 0.162 | 0.175 | 0.178 | 0.173 | 0.165 | 0.164 | 0.174 | 0.148 | 0.167 | 0.010 |
| to | 0.106 | 0.105 | 0.109 | 0.113 | 0.088 | 0.101 | 0.060 | 0.080 | 0.095 | 0.018 |
| . | 0.066 | 0.058 | 0.058 | 0.063 | 0.078 | 0.082 | 0.081 | 0.084 | 0.071 | 0.011 |
| in | 0.091 | 0.097 | 0.085 | 0.069 | 0.065 | 0.054 | 0.058 | 0.069 | 0.074 | 0.016 |
| and | 0.050 | 0.044 | 0.045 | 0.050 | 0.106 | 0.082 | 0.081 | 0.077 | 0.067 | 0.023 |
| a | 0.072 | 0.065 | 0.064 | 0.059 | 0.059 | 0.080 | 0.033 | 0.053 | 0.061 | 0.014 |

The next step is to represent the author's profile using Z score values (an approach called profile-based). The basic results are reported in Table 3.3. For each author, the mean over all his articles is computed for each word (values shown in Table 3.2) and stores into a single vector (denoted $rtf$ with H or M in Table 3.3). For example, for

the determiner *the* in Hamilton's profile, the value 0.273 is the mean over the values 0.257, 0.271, 0.271, and 0.293. The relative term frequency vector for the disputed article (Q54) is also reported in Table 3.3. Finally, using the means and standard deviations shown in Table 3.2, the Z score values of all those terms are derived for both authors and reported in the next two columns of Table 3.3. For example, the conjunction *and* in Hamilton's profile has a relative term frequency of 0.047. To obtain its Z score, this value is standardized as $(0.047 - 0.067)/0.023 = -0.862$.[1]

**Table 3.3** Relative frequencies and Z scores of the eight MFWs (Hamilton and Madison)

| | *rtf* | | | Z score | | |
|---|---|---|---|---|---|---|
| Word | H | M | Q54 | H | M | Q54 |
| the | 0.273 | 0.258 | 0.281 | 0.303 | −0.303 | 0.638 |
| , | 0.188 | 0.212 | 0.202 | −0.691 | 0.691 | 0.126 |
| of | 0.172 | 0.163 | 0.160 | 0.478 | −0.478 | −0.721 |
| to | 0.108 | 0.082 | 0.084 | 0.722 | −0.722 | −0.653 |
| . | 0.061 | 0.081 | 0.081 | −0.894 | 0.894 | 0.854 |
| in | 0.085 | 0.062 | 0.091 | 0.759 | −0.759 | 1.083 |
| and | 0.047 | 0.086 | 0.053 | −0.862 | 0.862 | −0.614 |
| a | 0.065 | 0.056 | 0.049 | 0.314 | −0.314 | −0.842 |

The components included in both profiles can easily be interpreted. A negative value indicates that this writer tends to employ this lexical item less frequently than the mean. For Hamilton, the *in* exposes a positive Z score of 0.759, indicating that this author uses this preposition more often. For Madison, the Z score for *the* is −0.303 indicating that Madison employs this determiner less frequently than the mean. The word-types *to* and *a* depict the same picture, both are used more regularly by Hamilton.

Once these Z score dimensionless quantities are obtained for each stylistic feature, they can be compared to those obtained from other texts or author profiles. In the current case, Table 3.3 reports in its last column the Z score computed with the questioned article Q54.

From the Z score vectors, a distance value between pairs of texts can be computed. Given a query text $Q$, an author profile $A_j$, and a set of terms $t_i$, for $i = 1, 2, \ldots, m$, the Delta value (or the distance) is evaluated by applying Eq. 3.2.

$$Delta(A_j, Q) = \frac{1}{m} \cdot \sum_{i=1}^{m} |Z\ score(t_{i,A_j}) - Z\ score(t_{i,Q})| \tag{3.2}$$

[1] My computation is based on a more precise representation than the rounded values depicted in Table 3.2. You can verify this with the Excel sheet available in the dedicated webpage.

In this distance computation, the same importance is attached to each term $t_i$, independent of their absolute occurrence frequencies. The impact of each term depends only on their Z score values. When inspecting data reported in Table 3.3, clearly the Z score value associated with *the* under Hamilton's profile (0.303) is similar to the value associated with *a* (0.314) even though the determiner *the* appears more frequently than *a* (absolute frequency of 763 vs.181 in the four papers written by Hamilton).

In the Delta computation, large differences occur when, for an item, both Z scores are large and have opposite signs. In this case, one author tends to use this term more frequently than the mean, while the other employs it very infrequently. However, when for all terms the Z scores are very similar, the distance value between the two style representations would be small, indicating the same author had probably written both texts ("less unlike" wrote Burrows [46]). A deeper mathematical explanation of Eq. 3.2 can be found in [12, 111].

In our example, the $Delta$ (Hamilton, Q54) = 0.900 and $Delta$ (Madison, Q54) = 0.713, indicating that the query Article Q54 could have been written by Madison (a correct decision). But as mentioned previously, the number of features ($m = 8$) is too small to obtain a reliable and robust attribution scheme.

Instead of applying a profile-based approach generating a unique vector for each author, an instance-based model can be adopted. In this case, each article is represented by a Z score vector. To achieve this, each relative frequency appearing in Table 3.2 is standardized (by applying Eq. 3.1). The resulting vectors are reported in Table 3.4.

**Table 3.4** Z scores of the eight articles written by Hamilton and Madison

| Word | H59 | H60 | H61 | H65 | M37 | M38 | M47 | M48 |
|---|---|---|---|---|---|---|---|---|
| the | −0.321 | 0.205 | 0.221 | 1.105 | −1.055 | −1.241 | 1.625 | −0.539 |
| , | −0.239 | −0.786 | −0.611 | −1.126 | 0.086 | 0.184 | 0.370 | 2.123 |
| of | −0.560 | 0.773 | 1.090 | 0.608 | −0.278 | −0.359 | 0.689 | −1.962 |
| to | 0.589 | 0.554 | 0.767 | 0.979 | −0.404 | 0.319 | −1.950 | −0.855 |
| . | −0.454 | −1.229 | −1.170 | −0.723 | 0.567 | 0.953 | 0.872 | 1.184 |
| in | 1.124 | 1.476 | 0.759 | −0.321 | −0.549 | −1.250 | −0.971 | −0.268 |
| and | −0.743 | −1.005 | −0.944 | −0.755 | 1.719 | 0.662 | 0.622 | 0.443 |
| a | 0.823 | 0.310 | 0.221 | −0.100 | −0.131 | 1.400 | −1.970 | −0.553 |

Then for each Z score vector, a distance value is computed with the disputed text Q54 according to Eq. 3.2. In our example, the smallest distance is achieved by $Delta(H59, Q54) = 0.734$ and the second smallest by $Delta(M34, Q54) = 0.924$. This is an indication that the possible true author of Article Q54 is Hamilton (an incorrect attribution).

Several variants of the Delta method have been suggested. The resulting effectiveness is usually similar to the classic Delta indicating that this model can be considered as a robust test for authorship attribution across different text genres

and natural languages [331]. For example, as a possible variation, one can discuss the contracted forms (e.g., *don't*, *I'll*, etc.). One can argue that these forms closely reflect an author's choice, while others suggest to expand all those forms because they might reflect more an editor's decision than a real author's choice. As a third option, Hoover [169] proposes to remove them.

As another alternative, Hoover [168, 169] proposes to ignore all personal pronouns from the feature set because they could represent more noise than useful stylistic markers. In fact, the frequency of these pronouns is related to the text genre, with a higher density of them in dialogues (e.g., in plays) and a lower one in narrative novels. The text genre has a real impact on the stylistic representation.

When defining the most frequent word-types, one can remove terms occurring with a very high frequency in one text (*culling process*). For example, Hoover [169] proposes to ignore words for which one text supplies more than 70% of all occurrences. A typical example is the presence of personal names inside a novel. These word-types are very frequent but only in a single text. Obviously, one can modify this threshold of 70% with a lower or higher percentage, together with the limit of a single text, by considering a few documents.

Equation 3.2 can also be slightly modified, for example, by ignoring absolute differences smaller than 0.3 (or another threshold) [168, 169]. The idea is to take account of some variability in these stylistic measurements. An author will not write texts with exactly the same percentage for all functional items. Assuming some differences within the style of a given person seems a reasonable choice. Thus, small Z score value differences should be ignored. As a complementary variant, the distance is computed only for differences presenting Z scores with opposite signs.

As other possible variants, the distance computation shown in Eq. 3.2 could be replaced by another function such as the Cosine (see Sect. 3.4). Finally, with the *rolling Delta* [99, 333], this model could be applied to detect collaborative work (see Sect. 7.5).

## 3.2 Kullback–Leibler Divergence Method

Representing the specific style of an author could signify counting the relative frequencies of the $m$ most frequent word-types. But if the very frequent word-types of a given language correspond to the functional words, it is not required to define them according to a corpus. One can simply define them prior to any investigation.

Adopting this viewpoint, Zhao and Zobel [431] suggest considering a limited number of predefined word-types to discriminate between several authors. They proposed a fixed English wordlist that contains 363 terms covering mainly functional ones (e.g., the, in, but, not, am, of, can), and also certain frequently occurring forms (e.g., became, nothing). Other entries in this wordlist are not very frequent (e.g.,

howbeit, whereafter, whereupon), while some reveal the underlying tokenizer's[2] expected behavior (e.g., doesn, weren) or seem to correspond to certain arbitrary decisions (e.g., indicate, missing, specifying, seemed). As an alternative, one can apply Antonia's et al. [11] list containing 192 entries corresponding to functional words occurring in Early Modern English and Victorian English (e.g., the obsolete form *thou* appears in this list).

After defining this fixed feature set, the probability of occurrence of each term or feature for an author profile or a disputed text must be estimated. Based on these estimations, the degree of disagreement between the two probabilistic distributions can be evaluated. To achieve this, Zhao and Zobel [431] proposed using the Kullback–Leibler divergence (KLD) formula, also called *relative entropy* [107, 249]. The KLD value is expressed in Eq. 3.3 and indicates how far the feature distribution derived from the query text $Q$ diverges from the $j$th author profile distribution $A_j$.

$$KLD(Q||A_j) = \sum_{i=1}^{m} p(t_i, Q) \cdot \log_2 \left( \frac{p(t_i, Q)}{p(t_i, A_j)} \right) \tag{3.3}$$

where $p(t_i, Q)$ and $p(t_i, A_j)$ indicate the occurrence probability of the term $t_i$ in the questioned text $Q$ or in the $j$th author profile, respectively. In this computation, it is assumed that $0 \cdot \log_2(0/p) = 0$, and $p \cdot \log_2(p/0) = \infty$.

With this definition, and when the two distributions are identical, the resulting value is zero, while in all other cases the returned value is positive. An example will clarify the underlying computation. In Table 3.5, three authors' profiles are reported, as well as the stylistic surrogate of the query text $Q$ based only on three terms. A quick look at the data depicted in this table shows that author $A_3$ is the closest to the disputed text, while author $A_1$ is the farthest away. The style of author $A_2$ represents a uniform distribution over the three terms.

**Table 3.5** Representation of three authors' profiles together with a query Text Q

| | Term $t_1$ | Term $t_2$ | Term $t_3$ |
|---|---|---|---|
| $A_1$ | 0.1 | 0.2 | 0.7 |
| $A_2$ | 0.333 | 0.333 | 0.333 |
| $A_3$ | 0.45 | 0.35 | 0.2 |
| $Q$ | 0.5 | 0.3 | 0.2 |

In Table 3.6, each cell indicates the contribution of each term according to each author's profile when considering the query Text $Q$. The overall score is reported in the last column of Table 3.6 (under the label "KLD"). When the estimated probabilities for a term are the same in the query text and in the profile, the impact

[2] A tokenizer is a program used to split a text into a sequence of tokens (word-type instances).

is nil. When this estimate is larger in the query text than in the author's profile, the computed value is positive (and negative in the opposite case).

**Table 3.6** Contribution of each component and final KLD value

| | Term $t_1$ | Term $t_2$ | Term $t_3$ | KLD |
|---|---|---|---|---|
| $KLD(Q\|\|A_1) =$ | 1.161 | 0.175 | −0.361 | 0.975 |
| $KLD(Q\|\|A_2) =$ | 0.293 | −0.046 | −0.147 | 0.100 |
| $KLD(Q\|\|A_3) =$ | 0.076 | −0.067 | 0.000 | 0.009 |

With this approach, the main concern is to *accurately estimate* the occurrence probabilities. As a first estimation for term $t_i$ (either $p(t_i, Q)$ or $p(t_i, A_j)$), the *maximum likelihood* principle can be applied. In this case, the relative frequency ($rtf_{i,j}$) shown in Eq. 3.4 respects this principle.

$$p(t_i, D_j) = rtf_{i,j} = \frac{tf_{i,j}}{n} \tag{3.4}$$

where $tf_{i,j}$ indicates the absolute term frequency (or the number of occurrences) of term $t_i$ in the $j$th document (or in a sample of texts written by author $A_j$), and $n$ the text or sample size (in number of tokens).

This first solution tends to overestimate the occurrence probability of terms appearing in the sample at the expense of the missing ones. Since the occurrence frequency for the latter is 0, its probability would also be 0, for example, when an author does not use a given word. However, the word distribution follows the LNRE law (*Large Number of Rare Events* [14]), whereby new words always tend to appear. Thus estimating them with 0 is not fully correct. To correct this, a smoothing must be applied, eliminating all probabilities equal to zero. As a side effect, any special processing resulting from an occurrence probability of 0 can be ignored.

As a first smoothing approach, Laplace suggests adding one to the numerator in Eq. 3.4 and likewise adding the vocabulary size to the denominator [249]. This approach could then be generalized by using a λ parameter (Lidstone's law [242]), resulting in the following probability estimates:

$$p(t_i, D_j) = \frac{tf_{i,j} + \lambda}{n + \lambda \cdot |V|} \tag{3.5}$$

with |V| indicating the vocabulary size.

It is not always clear how to fix the value for the parameter λ. Specifying $\lambda = 1$, the smoothing is equivalent to Laplace's method. Fixing a value smaller than 1 (e.g., 0.1) avoids assigning a relatively high probability to rare words, since in authorship attribution rare terms are usually not of prime importance. In certain circumstances, the maximum likelihood estimation (see Eq. 3.4) would be better [125], thus justifying a smaller value for the parameter λ.

As an alternative, Zhao and Zobel [431] suggest using the Dirichlet smoothing method which estimates occurrence probabilities by applying the following equation:

$$p(t_i, D_j) = \frac{tf_{i,j}}{n+\mu} + \frac{\mu}{n+\mu} \cdot p(t_i, B) \tag{3.6}$$

where $p(t_i, B)$ is the probability of term $t_i$ in a background model, and $\mu$ a parameter applied to adjust the importance of direct estimation versus that of the background model [250]. This smoothing scheme was found effective in information retrieval (IR).

With this smoothing approach, the resulting estimation relies on a mixture of direct estimation ($tf_{i,j}/(n+\mu)$) and probability provided by a background model B. This latter model is useful when the corresponding frequency $tf_{i,j}$ equals 0, or when the size $n$ of the sample is small, often resulting in inaccurate estimates. In such cases, the background model may provide better estimates of the probabilities. Opting for this scheme requires both the presence of a background corpus and defining an appropriate value for the parameter $\mu$ (in the range 0.001–10,000).

Adapting Zhao and Zobel's model for another language requires having a short wordlist of very frequent words appearing in that language. A simple solution is to select a stop wordlist available for many different languages[3] to identify author's style idiosyncrasies. Such a list is applied by search engines to avoid looking for terms without a pertinent meaning, and this is precisely one possible definition of functional words.

## 3.3 Labbé's Intertextual Distance

As with the previous authorship methods, the intertextual distance proposed by Labbé and Labbé [225, 229] begins by defining a set of terms and proposing a text representation based on these characteristics. Instead of selecting a subset of the vocabulary or only the functional terms, this model proposes to take account of the entire *lexis* or, in other words, of all word-types.

In addition, the author is not represented by a single profile built by concatenating all his writings. For each text (e.g., novel, play, set of tweets), a surrogate is generated with the corresponding $tf$ values and the label indicates the true author. Having $k$ documents with known authorship, the system is working with $k$ surrogates (instance-based).

The representation of the disputed text denoted $Q$ follows the same procedure. A surrogate is built based on the absolute term frequencies. Then the intertextual distance can be computed between $Q$ and all other text representations as described

[3] See, for example, http://www.members.unine.ch/jacques.savoy/clef/.

in Eq. 3.7.

$$D_{Labbé}(A, Q) = \frac{\sum_{i=1}^{m} \widehat{tf}_{i,A} - tf_{i,Q}}{2 \cdot n_Q} \quad \text{with } \widehat{tf}_{i,A} = tf_{i,A} \cdot \frac{n_Q}{n_A} \tag{3.7}$$

In this formulation, the distance between Text A and $Q$ is denoted $D_{Labbé}(A, Q)$. To achieve this, the absolute difference of all $m$ terms is taken into account. For each term, the difference between the absolute frequency in Text A and $Q$ is computed. At this point, an important problem arises. Both texts do not have similar lengths, and computing the difference in absolute frequency makes no sense.

To solve this issue, both texts must have the same length. Let us assume that Text A is larger than Text $Q$. Then all $tf$ values of terms belonging to $A$ are multiplied by the ratio of the two lengths (as shown in the right part of Eq. 3.7). If $n_A = 200$ and $n_Q$ to 100, each original $tf_{i,A}$ is divided by two. When summing all the modified $tf_{i,A}$ values (denoted $\widehat{tf}_{i,A}$), the text size for $A$ is $n_Q$.

With this approach, the denominator of Eq. 3.7 (namely $2 \cdot n_Q$) corresponds to the maximal distance between the two text surrogates. When the two texts are exactly the same, both $tf$ values are identical. Thus, for each term, the resulting difference is always zero, returning an intertextual distance of 0.0. However, when the two documents have nothing in common (e.g., one is written in English, the other in Chinese), when a $tf$ is positive for one text, it is nil for the other (and vice versa). Computing the sum for all term frequency differences, the numerator will be equal to the sum of the size of the two texts. This sum is also equal to the denominator. The distance then reaches its maximum value of 1.0.

The intertextual distance returns a value between 0.0 and 1.0 depending on the lexical overlap between two texts. More precisely, this distance value depends on both the number of terms appearing in the two documents and their occurrence frequencies. The returned distance value has a natural interpretation, which was not the case with the Delta and KLD approach.

A few additional comments must be taken into consideration. First, this model works well when faced with texts having a high quality, meaning no spelling errors. We must guarantee that each match is performed between the same lexical items. Second, all extra-textual items have been removed (e.g., text not reflecting the author style such as page numbers, chapter titles, etc.). Third, the text length must be larger than 5000 words [63], but we suggest to increase this limit to 10,000 words. A skilled writer can hide his own identity for a few hundred words (see Sect. 7.9), or imitate the style of another author, but it becomes more difficult to do so in a longer sample. Fourth, the length difference between the two texts must be smaller than 1:5. It is difficult to compare term frequencies between documents having a large length difference. When respecting these constraints, an intertextual distance of 0.2 or smaller is a very strong indication that both texts are written by the same author. Based on a training sample, this threshold can be verified assuming a distance value distribution [342].

This model was applied successfully in different applications, such as clustering political speeches [404], in the authorship dispute between R. Gary and E. Ajar [226], in literature [211], and to detect duplicate and automatically generated scientific articles [230], or in detecting fake nucleotide sequences [231].

## 3.4 Other Distance Functions

Based on two feature sets, both Burrows' Delta and Labbé's intertextual distance are calculated on the absolute differences of each selected term. For Zhao and Zobel, each selected item is represented by an occurrence probability estimation. Then the divergence between two probability distributions is computed by applying the Kullback–Leibler formula. In these three examples, the aggregation function is not fully justified and one can see this as an ad hoc approach. Such a strategy is the norm in stylistic studies due to the absence of a general theory justifying a precise distance measure.

For Delta and Labbé's method, the distance function is mainly based on the Manhattan formulation given in Eq. 3.8 in which the variable $a_i$ indicates the value of the $i$th term (or component) inside the vector A (and similarly for $b_i$ in vector B).[4] In this formula, $m$ indicates the number of components of the vector A or B. In our context, $m$ indicates the number of stylistic features (e.g., the number of word-types in Burrows' Delta).

$$D_{Manhattan}(A, B) = \sum_{i=1}^{m} |a_i - b_i| \tag{3.8}$$

In this expression, only the amplitude of absolute difference is taken into account. Distance functions based on this characteristic are derived from the $L^1$ norm (e.g., the Delta model or Labbé's distance). This distance value is equal to 0 when both vectors are identical and positive in all other cases. The maximum distance value is unknown, rendering the interpretation of the distance value problematic. To avoid this problem, Labbé and Labbé [229] suggest normalizing the distance by the maximum distance (see Eq. 3.7, and other variants are provided in Sect. 6.1).

Distance functions based on the $L^2$ norm have another starting point; large differences must account for more than small ones when computing an overall distance value. One solution to satisfy this desideratum is to consider applying the Euclidian distance depicted in Eq. 3.9. Taking the power of two for each difference increases the impact of large ones and reduces the importance of small ones.

[4] It is usually assumed that each component of a vector is positive or null.

$$D_{Euclidian}(A, B) = \sqrt{\sum_{i=1}^{m}(a_i - b_i)^2} \tag{3.9}$$

As with the Manhattan, the returned value is positive but without having a precise maximum value. A meaningful interpretation of a value larger than 0 is therefore problematic. We are never sure if such a value is really small, rather small, in the mean, large, or huge. A comparative basis is mandatory.

As a third paradigm, one can consider a distance measure based on the *inner product*[5] also called dot product or scalar product. This distance is provided by Eq. 3.10.

$$D_{InnerProduct}(A, B) = \sum_{i=1}^{m} a_i \cdot b_i \tag{3.10}$$

In this family, the most popular distance measure in computer science is the cosine function. Equation 3.11 depicts the computation of this *similarity* measure (this is not a distance!). The numerator of this formula corresponds to the inner product between the two vectors A and B. In the denominator, one can see the product of the two (Euclidian) norms[6] (or vector lengths).

$$Sim_{Cosine}(A, B) = \frac{\sum_{i=1}^{m} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{m} a_i^2} \cdot \sqrt{\sum_{i=1}^{m} b_i^2}} \tag{3.11}$$

The value of this measure is limited to the range [0–1], leading to a clear interpretation. When the two vectors have nothing in common, the numerator reaches the value 0, which is also the returned value. In contrast, when the vector A is equal to B, the numerator is the sum of all the components, at the power of 2. In this case, the denominator is the same, two times the square root of the same value. Therefore, the final value is 1. If the similarity is 100%, the two vectors are identical.

When using the cosine as a distance metric, one needs to transform this similarity value into a distance value as shown in Eq. 3.12. This distance function returns values between 0 and 1, with 1 indicating the largest distance.

$$D_{Cosine}(A, B) = 1 - Sim_{Cosine}(A, B) \tag{3.12}$$

[5]The inner product is more general [88, Section 3.2], but for simplicity, only this definition will be used for the moment.

[6]The reader can see that the norm of a vector is the square root of the inner product (or dot product) of the vector with itself.

As a fourth paradigm, an entropy-based distance function can be proposed, for example, the KLD formula (see Eq. 3.3). Clearly, this is not a metric because the KLD expression does not respect the symmetry constraint.

In conclusion, several distance functions can be applied and, for each of them, different variants can be implemented. With the Delta model, for example, Evert et al. [111] describe and explain several variations (e.g., applying a Euclidian, or a cosine distance function) within the framework proposed by Burrows [46]. As another example, Hoover [169] also suggests using the cosine distance with the Delta model (without achieving significantly better effectiveness).

Moreover, when selecting a distance measure to compute the distance between two stylistic representations, theoretical justifications do not provide a clear guide. Based on the $L^1$ norm, the Manhattan function applied in the Delta model and Labbé's intertextual distance is simple to compute. From a performance point of view, Kocher and Savoy [209] demonstrate that some variants of the $L^1$ family (e.g., Tanimoto, see Sect. 6.1) tend to provide higher effectiveness in several author profiling tasks. However, a complete answer to the question "Which is the best distance function for a given stylistic task?" remains elusive. This issue is related to the *no free lunch theorem* [422, 423]. When averaging the effectiveness over all possible problems, every classification algorithm has a similar accuracy rate when classifying new unseen data. In other words, no learning scheme can be universally better than all the others.

## 3.5 Principal Component Analysis (PCA)

In the stylometry community, the Principal Component Analysis (PCA) has been applied for its capability to visualize, in a two-dimensional figure, the stylistic representations of texts or text samples. A good introduction to the geometric and algebraic aspects of the PCA model can be found in the seminal paper of Binongo and Smith [34]. Concrete application examples can be found in [16, 92, 161, 172, 204], or [407]. The main objective of this section, therefore, is to briefly illustrate the interest of PCA for authorship identification.

Coming back to our *Federalist Papers* problem. Let us simplify it by considering only four lexical items (*upon*, *on*, *by*, *would*) with respect to four articles written by Hamilton (H59, H60, H61, and H65), four by Madison (M37, M38, M47, and M48), and two disputed texts (Q49, Q54). The absolute occurrence frequency of these four features according to these ten articles is reported in Table 3.7. The last two columns indicate the mean and standard deviation ($sd$) of each lexical item over the ten texts.

Evidently, the preposition *upon* is used more frequently by Hamilton while *on* appears more often with Madison. Usually a high frequency of *by* corresponds more to Madison's style and the verb *would* is also employed more regularly by Hamilton.

When analyzing the data reported in Table 3.7 three difficulties appear. First, the text length of each article is certainly not the same but could be considered as similar

**Table 3.7** Absolute frequency of some selected words (Hamilton and Madison)

| Word | H59 | H60 | H61 | H65 | M37 | M38 | M47 | M48 | Q49 | Q54 | *mean* | *sd* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| upon | 3 | 8 | 3 | 10 | 1 | 4 | 0 | 0 | 0 | 2 | 3.1 | 3.45 |
| on | 6 | 6 | 6 | 5 | 19 | 15 | 20 | 16 | 16 | 19 | 12.8 | 6.27 |
| by | 17 | 21 | 5 | 16 | 30 | 37 | 42 | 28 | 15 | 26 | 23.7 | 11.12 |
| would | 16 | 28 | 17 | 25 | 7 | 15 | 4 | 3 | 22 | 6 | 14.3 | 8.99 |

(all are newspaper articles). Second, some items appear with a higher frequency (e.g., the preposition *by* with Madison) compared to others (e.g., *upon*). Third, each vector has four dimensions (one for each feature or word-type), and thus it is not possible to represent them into a two-dimensional graph.

**Table 3.8** Centered frequency of some selected words (Hamilton and Madison)

| Word | H59 | H60 | H61 | H65 | M37 | M38 | M47 | M48 | Q49 | Q54 |
|---|---|---|---|---|---|---|---|---|---|---|
| upon | −0.1 | 4.9 | −0.1 | 6.9 | −2.1 | 0.9 | −3.1 | −3.1 | −3.1 | −1.1 |
| on | −6.8 | −6.8 | −6.8 | −7.8 | 6.2 | 2.2 | 7.2 | 3.2 | 3.2 | 6.2 |
| by | −6.7 | −2.7 | −18.7 | −7.7 | 6.3 | 13.3 | 18.3 | 4.3 | −8.7 | 2.3 |
| would | 1.7 | 13.7 | 2.7 | 10.7 | −7.3 | 0.7 | −10.3 | −11.3 | 7.7 | −8.3 |

Instead of using the raw data directly as shown in Table 3.7, we subtract from each frequency its mean (e.g., the mean over the ten observations). These centered values are reported in Table 3.8. For example, for the preposition *upon* in H59, the centered value is $3 - 3.1 = -0.1$. These values indicate when an author used a certain word-type more frequently (resulting in a positive value) or less frequently (negative value) than the mean. Selecting only the first two lexical items (*upon* and *on*) in Table 3.8, the different stylistic representations can be displayed on a graph as reported in Fig. 3.1.

When a newspaper article presents a large number of *on*, it will appear on the right and texts where *upon* occurs many times occur on the top. In Fig. 3.1, texts written by Madison are depicted on the right and those authored by Hamilton on the top-left corner. The two questioned documents (Q54 and Q49) appear close to Madison's texts. This might be viewed as an indication (based only on two lexical items however) that Madison might be the true author of these two disputed articles. One can see that Madison's article M48 and disputed text Q49 have the same coordinates (exactly the same difference to the mean for the prepositions *on* and *upon*).

Instead of being limited to two features, all four lexical items (or variables) could be used to indicate the position of each article (in a four-dimensional space). As it is impossible to visualize such a space in two dimensions, PCA projects the points (defined in four dimensions) into a two-dimensional graph. To achieve this, the main idea is the following. PCA generates a first axis (called first *principal component*) defined as a *linear combination* of the four variables and respecting, as

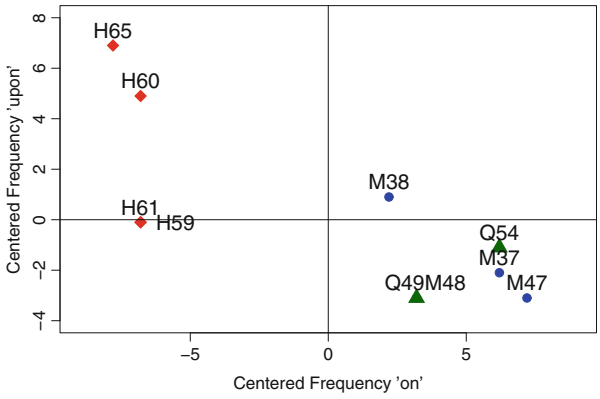


**Fig. 3.1** Centered frequencies of *on* and *upon*

best as possible, the real Euclidean distances between the points (computed using the four dimensions). This aspect is achieved by taking account of the standard deviation (or variance) of each lexical item. According to data depicted in Table3.7, the preposition *by* has the largest standard deviation, followed by the verb *would* and the preposition *on*. These three lexical items will have a large impact on the first axis because they represent the largest variability (or distance) inside the sample of ten articles.

After determining the first axis, PCA generates a second axis (the second principal component), always a linear combination of all variables (features), by maximizing the real distance between points. In addition, this second axis must be orthogonal to the previous one. Then, a third axis can be defined in a similar way, etc. up to the number of variables (four in our example). Each axis (principal component) represents a fraction of the total variability occurring in the sample. The order of the principal components is defined by taking a decreasing part of the total variability.

To visualize the relative position of the ten articles, one can select the first two principal components as shown in Fig. 3.2. To read it, the percentage of the total variability is indicated for each axis. In this example, the first axis (horizontal line) represents 74.3% of the total variability, while the second axis (vertical) corresponds to 17.3%. Thus, this graph shows $74.3\% + 17.3\% = 91.7\%$ of the real distance between points. Because this is a toy-size example limited to four variables, the first two components explain a large proportion of the underlying variability. In more realistic cases, the percentage explained by the first two components will certainly be lower than in our example.

In Fig. 3.2, the texts written by Madison appear as blue rounds on the left part. Articles authored by Hamilton are depicted by red diamonds on the right. Query article Q54 occurs clearly on the left, very close to Madison's Article #37. The second disputed article, Q49, appears upper, in the middle, and the closest article is Hamilton's Article #59.

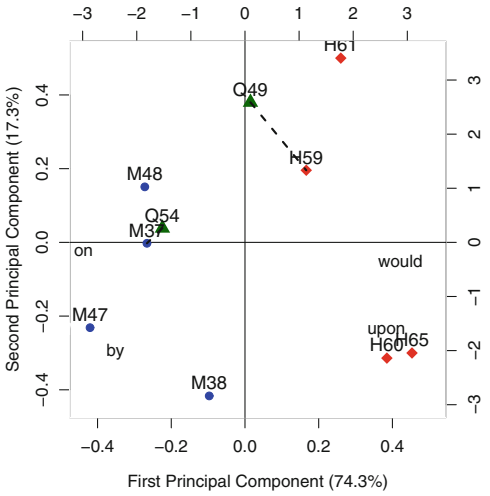


**Fig. 3.2** Principal Component Analysis (ten articles, four word-types)

To understand this graph, one can inspect the data reported in Table 3.8. The first axis is strongly related to the prepositions *by* and *on* and to the auxiliary verb *would*. Madison's texts contain more *by* and *on* than Hamilton's documents and the reverse for the verb *would*. Therefore, the first axis corresponds, in part, to texts having more occurrences of *on* and *by* than the mean, appearing on the left. Articles containing more *would* than the mean will occur on the right. To help understanding, the importance of each variable with respect to the two principal components is indicated in Fig. 3.2. For the second principal axis, the texts with a larger number of *by* and *upon* compared to the mean are located in the bottom part. On the top, one can see articles in which *by* occurs less frequently than the mean, for example, Q49 (−8.7, centered value reported in Table 3.8) and H61 (−18.7).

When applying a PCA, the user must specify whether or not the raw data must be standardized (as for the Delta model). Binongo and Smith [34] recommend the use of the standardized variables, particularly when these variables are not measured on a very similar scale on the one hand, or on the other, when their standard deviations vary greatly. In our example, as depicted in the last column of Table 3.7, the standard deviations are not the same across the four variables. Therefore, we had followed Binongo and Smith's recommendation.

Applying an additional step, one can compute the distance between projected points using the Euclidian distance function. Therefore, one can determine the closest document to a disputed text and thus define its probable author or category. In addition to having a visualization tool, PCA allows us to discover the true author of a query text or to determine its stylistic category (e.g., text genre, author age range, or gender, etc.). The closest distance between the two query articles and the others is depicted in dashed lines in Fig. 3.2.

**Table 3.9** Statements used to perform PCA in R with our example

```
> mydata <-read.table("DataTable3.7.txt", header=T, row.names=1)
> mydata
        H59  H60  H61  H65  M37  M38  M47  M48  Q49  Q54
upon      3    8    3   10    1    4    0    0    0    2
on        6    6    6    5   19   15   20   16   16   19
by       17   21    5   16   30   37   42   28   15   26
would    16   28   17   25    7   15    4    3   22    6

> options(digits=4)
> voc.pca <- prcomp(t(mydata), scale=T, center=T)
> summary(voc.pca)
Importance of components:
                          PC1    PC2    PC3    PC4
Standard deviation      1.724  0.832 0.4888 0.3126
Proportion of Variance  0.743  0.173 0.0597 0.0244
Cumulative Proportion   0.743  0.916 0.9756 1.0000

> voc.pca$rotation
                          PC1     PC2     PC3     PC4
  upon                 0.4796 -0.6248 0.25547  0.5606
  on                  -0.5469 -0.0626 -0.5342  0.6415
  by                  -0.4387 -0.7666 -0.0234 -0.4683
  would                0.5276 -0.1344 -0.8055 -0.2341

> voc.pca$x
                          PC1     PC2     PC3     PC4
  H59                 0.94356  0.5227 0.43398 -0.4742
  H60                 2.18540 -0.8388 -0.2784 -0.1417
  H61                 1.47581  1.3354 0.36967  0.0053
  H65                 2.57241 -0.8018 0.23422  0.3701
  M37                -1.51008 -0.0067 -0.0436  0.2175
  M38                -0.55054 -1.1128 -0.2114 -0.2070
  M47                -2.38607 -0.6181 0.04049 -0.2703
  M48                -1.54318  0.4023 0.50039 -0.0638
  Q49                 0.08436  1.0150 -1.1738 -0.0106
  Q54                -1.27170  0.1028 0.12851  0.5747
```

Table 3.9 reports the sequence of instructions required to perform a PCA on our small example using R. The first two statements are needed to read the data stored in the file `DataTable3.7.txt` and to display the data frame. Clearly, the occurrence frequency of four lexical items is provided for the ten articles. In the next step, the number of digits used to display a number is limited to 4.

The essential instruction is the call to the `prcomp()` function with the transpose of the data (because this function expects having the variables (*upon*, *on*, *by*, and

*would*) as columns). The other arguments specify to center and scale the data, or in other words, to standardize the data.

One then asks to present the results (`summary()`). The proportion of the variance explained by each principal component (PC) is provided (e.g., in the row "Proportion of Variance"). The values corresponding to the first two principal components and appearing in Fig. 3.2 are taken from this row. The next line indicates the cumulative proportion of the variation explained when considering one, two, three, or the first four principal components. As Fig. 3.2 is based on the first two principal components, 0.916 or 91.6% of the total variation is displayed.

With the command `voc.pca$rotation`, the linear combination of the variables forming the four principal components is displayed. One can verify that texts on the left of the graph will contain a larger number of *on* and *by*. For these two terms, the values under the column "PC1" is negative (e.g., −0.5469 and −0.4387).

The last instruction displays the coordinates of the ten articles according to the four principal components. The graphical representation in Fig. 3.2 only takes account of the first two (PC1 and PC2). For example, the point H60 has the coordinate (2.18, −0.839) and thus appears on the right and bottom part of the graph.

# Part II
# Advanced Models and Evaluation

The second part of this book corresponds to the core of this work, presenting in a detailed way a fair evaluation methodology, the numerous ways to generate stylistic features, and the different strategies to select the most pertinent ones. It then exposes the most important machine learning models to solve the authorship attribution or author profiling questions.

In this broad perspective, Chap. 4 starts by presenting some methodologies than can be applied to achieve a fair evaluation. Thus, it is the most mathematical chapter in this book. In fact, as soon as we are working with data to verify a hypothesis, the empirical evidence must be gathered with all the required rigor. Having a fair evaluation methodology is fundamental to verify whether one model performs statistically better than another or that their performance difference is not due to random effects. Moreover, evaluation is useful to show progress made over the years. Instead of generating new evaluation corpora, a list of possible test collections for both authorship attribution and author profiling is provided. They can serve as useful benchmarks to evaluate new models or variants of existing ones.

A second important aspect when conducting a stylometric application is to know all the possible manners to represent the writing style as discussed in Chap. 5. If isolated words are viewed as the prime source of evidence, one can also consider short sequences of words (or $n$-grams of words), or replacing words by their lemmas (or dictionary entries) or even by their stems. The decision is less clear when discussing punctuation symbols. In other cases, one can only include the part of speeches (POSs) (or grammatical categories) associated with each word or sequences of them. In addition, one can view the words as molecules and explore their constituent parts, the syllables, the letters, and short strings (or $n$-grams of characters). With social networks, other stylistic indicators could be considered, such as hyperlinks, hashtags, mentions, or the number of retweets. Of course, more complex representation can combine some of these stylistic indicators.

In most cases, the stylistic representation is grounded on the most frequent characteristics, or however, on those appearing often with one author (or category) and less frequently with the others. To identity the most pertinent stylistic markers, different feature selection functions have been proposed in data science. As several

procedures have been proposed without one clearly dominating the others, several approaches are presented with their advantages and drawbacks.

As shown in Chap. 6, several machine learning models can be applied to resolve different stylometric problems such as authorship attribution, and author profiling, or to identify some psychological traits of the author. With the $k$-NN ($k$ nearest neighbors) model, each text is represented by a point in a space formed by all selected stylistic features. When applying this method, one needs to define a distance (or similarity) measure between points to identify the $k$ closest ones to the document for which the authorship is uncertain. As a second well-known approach, the naïve Bayes method estimates the probability of occurrence of each feature according to a set of possible hypotheses (e.g., each of them corresponding to a possible author, or a stylistic category such as man or woman). More complex learning models could be applied to solve the authorship question such as the support vector machines (SVM). In this case, the method determines a border splitting the feature space into two distinct regions, one for each possible category (e.g., one reflecting the masculine writing style, the other the feminine one). As an alternative, one can consider the logistic regression model in which the probability of belonging to a target category is estimated based on a set of chosen predictors or features. To illustrate the application of these four machine learning methods, examples, written in R, are given and commented.

Chapter 7 focuses on some more recent stylometric models proposed either by the humanities or by computer scientists. In this overview, the Zeta test was proposed in a way to complement the Delta model (described in Chap. 3) by considering words or expressions used frequently by one author (or category) and mainly ignored by the second. Compression methods have also been proposed to discover the true author of a doubtful text. In this view, compressing the second text of a given writer should generate a smaller compressed file than when the two texts are authored by two distinct persons.

Specific approaches have been suggested to solve the verification problem. In this case, the investigator has only a sample of texts written by a unique author and a document of uncertain authorship. The central question is to know whether or not this query text was written by this single author. Chapter 7 describes several inventive strategies that have been proposed to solve this central issue.

Another important question in the stylometry domain is to verify a possible collaborative authorship within a given document. The collaboration between two or more authors is considered a hard problem, and a few strategies have been suggested to solve this question.

Finally, neural networks have a long history and some classical applications in stylometry are discussed in this last chapter. After presenting the basic components of a neural network, the word embeddings approach is exposed as a possible strategy to discover the true author of a document. As a second deep learning model, the recurrent neural networks and in particular the long short-term memory (LSTM) is described. After the significant successes encountered by deep learning models in image and handwriting recognition as well as in machine translation, one might expect similar performance in stylometric applications.

# Chapter 4
# Evaluation Methodology and Test Corpora


The organization of this chapter is as follows. Section 4.1 presents some preliminary comments about evaluation while the next describes some issues about text quality and its preprocessing. In Sect. 4.3, the main performance measures together with their advantages and drawbacks are explained. Possible variants are also briefly depicted. Section 4.5 briefly exposes the construction of confidence intervals. Section 4.6 introduces statistical tests able to detect whether or not a performance difference can be viewed as significant. Section 4.7 describes the problem of having a distinct training and test set. The next section exposes some classical authorship attribution problems (e.g., Shakespeare, St. Paul, the *Book of Mormon*). Section 4.9 briefly describes some of the PAN-CLEF test collections built from 2009 and covers all the stylistic problems described in the first chapter. Finally, and based on the *Federalist Papers* corpus, the evaluation of the stylistic methods presented in the previous chapter is performed and discussed.

In Sects. 4.5 and 4.6, not all the details of statistical procedures and inference are provided, topics covered by several introductory books in statistics such as [66, 82, 90, 139] (focusing more on quantitative linguistics), or [93] (dedicated for natural language processing applications). For readers not having any knowledge in statistics, Spiegelhalter's book [370] provides a gentle introduction to these topics and initiates the needed know-how.

## 4.1 Preliminary Remarks

To solve the authorship attribution question, a large number of authorship attribution methods have been proposed. For example, Rudman [325] indicates that more than 1000 feature sets are possible to solve the authorship attribution problem, without considering the number of possible classifiers and their parameters (see also [188]). The solution space is therefore rather large and arguments must be put forward



J. Savoy, *Machine Learning Methods for Stylometry*,
https://doi.org/10.1007/978-3-030-53360-1_4

to hopefully favor useful regions to be explored in priority. Theoretical arguments could favor various areas, but empirical evidence is also of prime importance. A suitable stylometric model must have both a solid linguistics justification and high performance. Under this second criterion, one can consider efficiency as, for example, the time needed to compute the answer or the amount of required storage. The efficiency is certainly important in a commercial service where the customer is impatient to obtain an answer. However, the major interest is related to another performance measure: the quality of the response or its *effectiveness*. This section describes this second aspect since efficiency is usually simpler to ascertain (e.g., time to execute a task, number of items in a ranked list to consult, number of keystrokes to be pressed, etc.).

A fair evaluation methodology is required to either measure the effectiveness of a model or rank several classifiers according to their performance. As another result, evaluation allows us to measure the progress made over time. As the stylometric community regroups scholars with distinct backgrounds, the evaluation principles might diverge between humanities and computer scientists or between arts and sciences. The former group places more importance on qualitative argumentation, while the latter favors quantitative results together with statistical tests able to detect significant effects. The main difference might reside in the response to the question "Can one trust the erudition of a scholar or the educated guess of a privileged expert who knows how to recognize the authentic style of an author." The answer is not binary (yes/no) because in the real world, a continuum exists between considering only qualitative to only quantitative arguments.

Authoritative arguments or justifications should not be considered as sound and valid (e.g., "disclosing that this novel with such a mediocre style cannot be written by that famous author"). For example, Hoover [167] asked his students to rank 12 novels (limited to the first 50,000 words) according to their vocabulary richness. In this experiment, the first two novels depicting the lowest lexical diversity (or the lowest number of distinct words) were always perceived as having the richest vocabulary. The students were confused between the reading difficulty and the vocabulary richness of a text, correlating strongly both measures indicating distinct stylistic aspects. As another example, Hoover [170] demonstrates that the unfinished work *Blind Love* (1890), left after the death of W. Collin, was finished by Besant. The latter was unable to hide his own style and imitate Collin's style.

Another important remark must be made when discussing evaluation. In humanities, interesting authorship questions or stylistic issues are usually limited to one or a few texts. For example, can one discover who is the secret hand behind Ferrante's novels, or whether or not Juliet's parlance truly corresponds to a female figure? Based on a single (or a few examples), it is not possible to honestly determine the accuracy rate of a proposed stylometric model or to compare the performance achieved by two (or more) classifiers.

One needs a test collection of texts to judge the effectiveness of one or more models. When generating such a test corpus [102], one must be absolutely certain that each document (or text excerpt) has the correct label (e.g., the author's name, the author's gender, age range, period of writing, etc.). This certainty is not always

easy to achieve. For example, to present the entire works of Shakespeare (the so-called Shakespeare's canon), one must determine which texts (or acts inside plays) have really been written by Shakespeare [390, 415]. If this concern seems more related to texts authored in the sixteenth or seventeenth century, one can suspect a collaboration between two authors or modifications and additions done by the editor (e.g., especially for works published after the author's death). When considering web data (e.g., chat, tweets, blogs, etc.), assuming that each pseudonym corresponds to a distinct person is certainly not always true. Some verifications can be done semi-automatically or manually when the corpus size is not too large.

Finally, to be useful, such a test collection must contain numerous examples, maybe at least 50 or better 100 instances of problems to be classified. These numbers represent the minimum value for applying, with some confidence, statistical procedures and tests. Of course, a larger number of instances are better and will allow for a more precise performance measurement.

## 4.2 Text Quality and Preprocessing

Data quality matters. "Yes, of course" could add the reader, "we know that." With computer systems, one can assume that the data error rate must be low or even very low. Different studies however indicate that the error rate on information systems or on a spreadsheet is not marginal (e.g., let us say around 1 or 2%). Such an error rate could reach a higher value, such as 88% [20, 293]. Strong et al. [382] found that 50–80% of computerized criminal records in the United States were inaccurate, incomplete, or ambiguous. Working with texts instead of numbers, a similar error rate could be expected in some circumstances. It is always a surprise to see that many published scientific articles tend to ignore this difficulty and do not describe any text preparation procedures. Preprocessing textual data is the norm, not an exception. This issue is not new and some previous studies have underlined the importance of this question [326, 375].

The spelling or more generally the orthography is the first issue. Working with an international language, linguists know that the spelling can change between countries (or even between smaller regions), such as the difference between the UK and US spelling [84]. The same difficulty appears in French or German, the latter have been the objects of various orthographic reforms during the past 40 years without achieving a global acceptance. Even when considering a language inside a single country, several variants of the same word-type could occur. For example, the Italian word *persino* (even) can be spelled as *perfino*. One author could choose one spelling or the other or change from one spelling to the next inside the same work.

Texts written in the sixteenth or seventeenth century also exhibit a large spelling variability. At that time, spelling did not respect a strict norm (given by an authoritative dictionary or according to a committee such as for the French language with the *Académie*). Spelling variations could also occur with copyists

and compositors.[1] For example, in their study of Shakespearian plays, Craig and Kinney [65] indicate that they conflate under the same entry in different variations (e.g., *folly*, *follie*, or *folie*). Some researchers suggest to conflate all these variations under the same form.

Web data, blog posts, tweets, or customer reviews present many spelling errors. Moreover, various smileys and emojis could indicate the same emotions or sentiments (e.g., :-) or :) or 😃). Do we regroup under the same form all possible spelling variants occurring in a corpus?

The transcription of foreign names opens the possibility of observing many spelling variants. Around 80 forms have been encountered to indicate Muammar Gaddafi (e.g., "El Kadhafi, Moammar," "Khadafy, Moammar," "Al Qathafi, Mu'ammar," etc.) The same issue can be found with some geographical or other entity names (e.g., "Al Qaida", "Al-Qaïda", "Al-Quaeda"). Older texts might reveal some variability for names, and for example, five distinct signatures of Shakespeare have been found, namely *Shakp*, *Shaksper*, *Shakspe*, *Shakspere*, and *Shakspeare*, and in later publications as *Shakespear*.

Various other inquiries must be clearly answered during the preprocessing stage. Some words could appear with their first letter in uppercase (capitalization) because they start the sentence or because this word corresponds to a proper name. In web data, words in tweets could appear in all uppercase letters (to underline their importance or to signal the writer speaks aloud). Do we need to transform all uppercase letters into their lowercase equivalent? Certainly, when an uppercase letter indicates the beginning of a sentence. For other occurrences, the answer could depend on the text genre and the target application.

And the punctuation symbols? And the numbers, dates, symbols, references, hashtags, hyperlinks, emoticons, and emojis, etc. It is not possible to provide rules that can be applied in all cases. The responses depend on the applications. Personally, I suggest to consider the main punctuation symbols (.,;?!) and to ignore less frequent ones (e.g., parenthesis), as well as numbers, dates, and other symbols ($, +). When working with web data, the specific features of a communication media (e.g., hashtags with tweet data) could be useful to discriminate between the underlying categories.

And the parts of speech? As mentioned previously with the Delta model, Burrows [46] takes care to distinguish between possible homographs, for example, with the preposition *to*, used also to indicate the infinitive (e.g., Ann goes *to* Glasgow *to* see Mary). Such a work could be delegated to a POS tagger [398], but the produced tagged corpus is certainly not error free. The result of a POS tagger is not always perfect on the one hand, and on the other, the processing can encounter ambiguities. For example, in the sentence "Time flies like an arrow," it seems reasonable to specify that "*time*" is the noun, subject of the sentence, and "*flies*" is the verb. Knowing that "*fruit flies*" are a kind of insect, the computer can infer

[1] For the compositors, the availability of some characters could explain the resulting spelling.

that "*time flies*" indicates a new species. This could then correspond to the subject of the sentence, and the verb is "*like*." Finally, only young programmers think they can program without bugs. As I become older, I am reminded more and more that error-free programming is a difficult task.

In addition to normalizing and correcting the text, the researcher must carefully inspect the document. To reflect as closely as possible the style of an author, the text must be preprocessed to remove tables, lists, guide words (running titles), chapter titles, page numbers, citations, hyphenation and dashes, apostrophes (running dialogue), quotations, etc. [168]. When working with plays, certainly the scenic indications (e.g., Romeo leaves the room) must be ignored too. "Good research requires that the data subjected to analysis should have been assembled with the highest scholarly care" specified Love [246, p. 154]). Moreover, other passages must be removed before the analysis such as parts of the play where the actor must speak another language (e.g., in pseudo-Turk in the play *Le Bourgeois Gentilhomme*).

As important and often neglected factors having an essential impact on the overall effectiveness of the text categorization system, one must mention the text length [98, 247]. Taking a decision based on a few words is a hard problem. With a text length of 10,000 words, the assignment could be proposed with good certainty. Considering a smaller size reduces the confidence attached to the proposed attribution. However, even a long text could be problematic when its orthography (spelling, capitalization, hyphenation, and punctuation) is rather poor (e.g., a transcript obtained with a low-performing OCR system). For example, Zbib et al. [430] indicate that the word error rate (WER) could be rather low for the English language (e.g., around 2%), but with languages having less electronic resources the WER can reach 30%. Therefore, some statistics about the text length distribution should be provided, as well as a clear description of the preprocessing of the data.

## 4.3 Performance Measures

Stylistic models can be applied to solve various problems. The simplest case is represented by the authorship attribution question or when faced with binary decisions (e.g., Is this text written by a man or a woman? Did Shakespeare write this play?). The proposed attribution or category is simply right or wrong. The answer returned by the system can be classified into two disjoint classes, namely correct and wrong.

For this simplest case, a natural effectiveness measure is the accuracy rate which varies from 0.0 to 1.0 (or from 0 to 100%). This measure is defined as the ratio between the number of correct decisions over the total number of decisions to be taken (or instances) (denoted by $s$) as reported in Eq. 4.1. In a dual view, the error rate is defined as 1.0 − the accuracy rate.

$$Accuracy\ rate = \frac{Number\ of\ correct\ decisions}{s} \tag{4.1}$$

As a second case, the classifier must determine the correct answer over three or more *nominal* categories. For example, the task consists of discovering the geographical origin of the speaker with, as possible answers, UK, United States, Australia, and South Africa. These labels are nominal because there is no relationship between them. One cannot sort them from the smallest to the largest or assume that one is more important than the others.

As an effectiveness measure, the accuracy rate can be chosen. However, this value can be computed according to two distinct schemes. First, the *micro-averaging* principle assumes that one decision corresponds to one vote. When the classifier produces, for example, 180 correct assignments out of a grand total of 200, the resulting accuracy rate (micro-average) is $180/200 = 0.9$ or 90%. In stylistic studies, this method is the most frequently used to compute an average performance.

Second, the accuracy rate could be computed for each category, under the assumption that the same importance must be assigned to each of them. In this case, one category corresponding to one vote (*macro-average*). The overall accuracy rate is the mean of all categories. For example, obtaining an accuracy rate of 0.9 for the first category, 0.5 for the second, 0.6 for the third, and 0.8 for the last one, the macro-averaging accuracy rate is $(0.9 + 0.5 + 0.6 + 0.8)/4 = 0.7$, or 70%. When each category contains the same number of instances, both measures return the same value.

Both the micro- and macro-average measures can be applied. In the machine learning domain, the first one usually tends to produce better results because frequent categories are assigned more importance and are usually easier to predict. With more data, a frequent category (or author) might be more precisely defined or the classifier would have more training data to distinguish between this particular category and the others.

As a similar but not identical case, the classifier must return the correct answer over three or more *ordinal* categories. An important difference occurs with the previous nominal case. The ordinal categories can be ranked, for example, with the age range of the author from teenager (14–20), young adult (25–35), and older person (45 and more). As described previously, one can opt for the accuracy rate and identify the answer as correct or wrong.

As an alternative, one can consider that an error between two adjacent categories must be viewed as less severe than an incorrect assignment involving more distant classes. One can replace the category labels by integers reflecting the order, for example, 1, 2, 3 in our example. Predicting the third class when the instance belongs to the first, the error value is $3-1 = 2$, while returning the second label will produce an error value of $2 - 1 = 1$.

Instead of applying the same distance between categories, one can chose another scale such as 1, 5, 6, a scale used to deeply penalize some incorrect decisions. In our example, predicting an adult (class 2) or an elderly person (class 3) when the correct decision is a teenager (class 1) is viewed as largely incorrect and penalized in consequence (e.g., $5 - 1 = 4$ and $6 - 1 = 5$). However, the error between an adult and a senior is viewed as marginal (e.g., $6 - 5 = 1$).

With ordinal categories, the evaluation measure takes account of the difference between the classes indicated by the answer (denoted $a_i$) and the correct one (denoted $c_i$) for the $i$th instance. Having $s$ instances to classify, the mean overall effectiveness is computed according to Eq. 4.2.

$$Mean\ error = \frac{1}{s} \cdot \sum_{i=1}^{s} |a_i - c_i| \tag{4.2}$$

A perfect system will achieve a value of 0 (no error), and all strictly positive values indicate the amplitude of the error. This evaluation approach penalizes the incorrect assignment differently according to the class difference. Therefore, this measure could be qualified as finer than the simple accuracy rate.

As a third case, the classifier returns not a single decision but a ranked list of possible categories (or authors), from the most probable answer to the least credible one. As described previously, only a single category corresponds to the correct decision. In such a context, a perfect classification scheme will always return the correct answer in the first rank. Presenting the correct category in the second or in any other position must be penalized. To achieve this, the reciprocal rank (denoted RR) can be computed for each instance as shown in Eq. 4.3.

$$Reciprocal\ Rank\ (RR) = \frac{1}{rank(1st\ correct\ answer)} \tag{4.3}$$

This performance measure corresponds to the inverse of the rank of the first correct answer. When the system ranks the correct decision in the first position, this function returns 1.0. Ranking the correct decision in the second place, the evaluation value is $1/2 = 0.5$, or only $1/4 = 0.25$ when the first correct answer appears in the fourth rank. When faced with a sample of $s$ instances, the overall effectiveness of a classification scheme is simply the mean over these $s$ assignments.

This evaluation measure is well-known in information retrieval [416], for example to evaluate a known-item search strategy. In this approach, not returning the correct answer in the first rank is clearly penalized. When the correct assignment occurs in the second rank, the effectiveness drops to 50%, from 1.0 to 0.5. Appearing in the third rank, the performance decreases to 33% (from 1.0 to 0.333).

As a fourth case, the classification scheme can provide a probability estimate for the single proposed decision, or more generally, a probability estimate for each returned decision (ranked in decreasing order to their probabilities). Let us start with an example in which the classifier returns a single decision. As explained previously, one can evaluate this scheme using the simple accuracy rate. But when the correct decision is supported with a probability of 0.51, one can infer that a large uncertainty is associated with this assignment. However, when the probability reaches 0.99, the system indicates a strong belief in its assignment. Ignoring the probability estimations, both answers are viewed as correct and provide the same performance.

When probability estimates are available, one can use them to evaluate the effectiveness of a classifier. The evaluation procedure is based on the difference between the probability estimate for each proposed answer (denoted $p_i$) and the real probability (denoted $q_i$). The latter is simply 1.0 for the correct category, and 0.0 for all others. With an instance having $k$ possible answers, the overall effectiveness is computed according to Eq. 4.4 corresponding to the quadratic loss function (QLF).

$$QLF = \sum_{i=1}^{k} (p_i - q_i)^2 \tag{4.4}$$

For example, with an instance having $k = 3$ possible categories, the classifier returns the correct decision with a probability of 0.6, and for the two others, with probability 0.3 and 0.1. The quadratic loss function for this instance is the sum of each $(p_i - q_i)^2$, namely $(0.6-1)^2+(0.3-0)^2+(0.1-0)^2 = 0.16+0.09+0.01 = 0.26$.

Applying this measure, an ideal system will achieve a performance of 0.0. In this case, for each category, one can find $p_i = q_i$. Assuming that the correct answer corresponds to the $j$th category, the QLF value can be computed equivalently according to Eq. 4.5.

$$QLF = 1 - 2 \cdot p_j + \sum_{i=1}^{k} p_i^2 \tag{4.5}$$

As explained previously, the performance over a sample of $s$ instances is simply the mean over these $s$ examples. The best classifier will depict the smallest mean quadratic loss value.

As a fifth case, the evaluation measure could take account of the number of unanswered questions, or when the system's response is *I don't know*. This answer is not correct on the one hand, and on the other, not fully wrong. Thus, the binary evaluation leading to an accuracy rate is not possible. To consider answers with these three values, Peñas and Rodrigo [298] suggest the c@1 measurement. This evaluation measure takes into account both the number of correct and incorrect answers and the number of instances left unsolved. The exact formulation is given in Eq. 4.6.

$$c@1 = \frac{1}{s} \cdot \left( nc + \frac{nc}{s} \cdot nu \right) = \frac{nc}{s} \cdot \left( 1 + \frac{nu}{s} \right) \tag{4.6}$$

In this formulation, $s$ indicates the number of instances, $nc$ the number of correct answers, and $nu$ the number of unanswered instances [376].

The perfect system will achieve a value of $c@1 = 1.0$, while the worst classifier will achieve the minimal value of 0. For example, with $s = 100$, $nc = 80$, and $nu = 0$, the accuracy rate is $nc/s = 0.8$, and c@1 gives the same value. However, when 10 of the "incorrect" decisions are left without an answer ($nu = 10$), the

c@1 measure does not view them as fully wrong, and the c@1 increases to 0.88. An unanswered instance does not penalize the overall performance as does a wrong answer.

As a variant that can also take account of the answer "I don't know," one can attribute 1 point when the decision is correct, 0 point when it is incorrect, and 0.5 when the system reply is *I don't know*. To determine the quality of an attribution scheme, one can sum these values to define an overall merit score. A perfect classifier will achieve a performance value of $s$ (corresponding to the number of instances to be classified), while the worst will obtain a value equal to 0. Of course, an incorrect decision could be penalized more strongly, for example, with a penalty value of $-1$ or $-2$. Considering the values of our previous example, with $s = 100$ and $nc = 80$ ($nu = 0$), the merit of this classifier is $80 \cdot 1 - 20 \cdot 0 = 80$. When only 10 decisions are wrong and 10 are unanswered, the performance is $80 \cdot 1 - 10 \cdot 0.5 - 10 \cdot 0 = 85$.

## 4.4 Precision, Recall, and F1 Measurements

The effectiveness measures presented in the previous section are mainly adapted for the authorship attribution problem. In other stylometric applications, the measurement must be associated with the correct identification of the various categories, for example, in author profiling. The set of possible categories could be limited to two (e.g., is this text written by a man or a woman?) or to a few (e.g., age ranges of the writer).

Instead of computing a global accuracy rate, the performance can be measured with the precision and recall. To understand these evaluation measures, the results are displayed according to a contingency table as shown in Table 4.1. In this case, the focus is on a category $C_j$ (e.g., texts written by a woman). The four main cells are denoted by TP (or true positive), TN (true negative), FP (false positive), and FN (false negative). Each of these cells contains the number of assignments corresponding to the column and line labels.

**Table 4.1** Contingency table for category $C_j$

| Category $C_j$ | | True state of Nature | |
|---|---|---|---|
| | | Yes | No |
| Proposed by the classifier | Yes | TP | FP |
| | No | FN | TN |

The TP cell indicates the number of instances classified in the category $C_j$ and that really belong to that class. Similarly, the TN represents the number of examples not belonging to the class $C_j$ and correctly identified by the classifier. The accuracy

rate is simply the ratio between (TP + TN) and number of classified instances or, in other words, represented by (TP + TN)/(TP + TN + FP + FN).

Two types of errors could occur in a classification process. The FP cell counts the number of instances incorrectly judged as belonging to the category $C_j$ by the classifier. However, the FN corresponds to examples belonging to category $C_j$ but not identified as belonging by the classifier. Based on this notation, the precision and recall for the category $C_j$ can be computed according to Eqs. 4.7 and 4.8. These measures are defined between 0 and 1 (or 0 and 100%), and the higher the value, the better the effectiveness.

$$Precision = P = \frac{TP}{TP + FP} \tag{4.7}$$

$$Recall = R = \frac{TP}{TP + FN} \tag{4.8}$$

An effective classifier must present a high precision and a high recall. These two measures are however negatively correlated. As one increases, the other tends to decrease. But both are required to obtain pertinent information about the quality of a classification scheme. Why?

Look at Table 4.1. One can design a simple classifier achieving a 100% recall. This is the trivial acceptor always generating the same answer: "yes." In this case, the cell FN is zero (the classifier never says "no") and the resulting recall is then 1. However, one can design a classifier to assign only one (or a very few) instance with the label "yes." If this decision is correct, the cell FP is zero and the achieved precision is 1.

Having two performance values, it could be difficult to compare and rank two (or more) classifiers. The first one might have a higher precision but a lower recall. To solve this problem, one can apply the $F_1$ measure as depicted in Eq. 4.9 corresponding to the harmonic mean (or the reciprocal of the arithmetic mean) between the precision and recall. As for the precision and recall, the $F_1$ value is defined between 0 and 1, with 1 specifying a perfect system.

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} \qquad \text{or } F_1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} \tag{4.9}$$

The $F_1$ measure can be generalized as $F_\beta$ as indicated in Eq. 4.10.

$$F_\beta = \frac{(\beta^2 + 1) \cdot P \cdot R}{(\beta^2 \cdot P + R)} = \frac{(\beta^2 + 1) \cdot TP}{(\beta^2 + 1) \cdot TP + \beta^2 \cdot FN + FP} \tag{4.10}$$

In this notation, the parameter $\beta$ denotes the importance assigned to the recall. Fixing $\beta = 0$, only the precision is taken into account, while when $\beta = \infty$ the returned value is the recall. With $\beta = 1$, the same weight is given to the precision and the recall. This is often the case and reflects the situation in which the two error

types (FN and FP) are viewed as having the same importance. When the emphasis must be placed on the recall, one can increase the $\beta$ value, for example, with the $F_2$ measure penalizing four times more the FN errors than the FP ones. To favor a classifier with a high precision, one needs to decrease the importance attached to FN errors, and thus decrease the $\beta$ value.

When faced with $r$ classes, the previous computation returns the precision and recall values for a given category $C_j$ as shown in Eq. 4.11.

$$P_j = \frac{TP_j}{(TP_j + FP_j)} \qquad R_j = \frac{TP_j}{(TP_j + FN_j)} \tag{4.11}$$

To aggregate the $r$ precision and recall measures, one can apply the macro-averaging principle assigning the same importance to each class (as shown in Eq. 4.12).

$$P_{macro} = \frac{1}{r} \cdot \sum_{j=1}^{r} P_j \qquad R_{macro} = \frac{1}{r} \cdot \sum_{j=1}^{r} R_j \tag{4.12}$$

$$P_{micro} = \frac{\sum_{j=1}^{r} TP_j}{\sum_{j=1}^{r} (TP_j + FP_j)} \qquad R_{micro} = \frac{\sum_{j=1}^{r} TP_j}{\sum_{j=1}^{r} (TP_j + FN_j)} \tag{4.13}$$

When opting for the micro-averaging principle, each decision has the same worth and the overall precision and recall values are computed according to Eq. 4.13. In this view, the category having the largest number of instances has more weight than the others.

## 4.5 Confidence Interval

Describing the effectiveness of a text categorization system by an estimated accuracy rate corresponds to the first step. This single performance value will however change when considering another sample or when adding (or subtracting) a few instances to the original sample. The question that then arises is to quantify this underlying variability. Based on a statistical procedure, a confidence interval can specify the lower and upper bounds of possible variations of this estimated accuracy rate (or the error rate defined as $1 -$ the accuracy rate). Instead of speaking about the accuracy rate, the same question can be viewed as the estimation of a confidence interval for a proportion (namely the proportion of good answers indicated by the accuracy rate).

The real proportion is a fixed but unknown value denoted by $p$. With the sample in hand, an estimation of this real proportion can be achieved by applying Eq. 4.1. This estimation is denoted $\hat{p}$ and can vary depending on the sample used to compute it. Thus $\hat{p}$ is a random variable.

To compute these two limits defining the confidence interval for $p$, one must indicate the required coverage specified by $1-\alpha$. If an error-free confidence interval is requested, the answer will be in the range between 0 and 1 indicated by [0–1]. Such an interval is however not informative. Every value is possible.

Therefore, it is important to indicate the percentage to be covered by the confidence interval, with a higher percentage implying a larger interval. In practice, the coverage is usually provided for 90 or 95% (implying that $\alpha = 10\%$ or $\alpha = 5\%$). In textbooks, the computation of the two limits for the confidence interval of a proportion is specified by Eq. 4.14 in which $s$ indicates the sample size and $z_{\alpha/2}$ the $\alpha/2$ percentile of the normal distribution. Behind this equation, one can assume that $\hat{p}$ is a random variable following a normal distribution with an estimated standard deviation (or $\widehat{sd}$) given by Eq. 4.15. This assumption holds when $s$ is rather large (e.g., larger than 50 or 100).

$$\hat{p} \pm z_{\alpha/2} \cdot \frac{\hat{p} \cdot (1 - \hat{p})}{\sqrt{s}} \tag{4.14}$$

$$\widehat{sd} = \frac{\hat{p} \cdot (1 - \hat{p})}{\sqrt{s}} \tag{4.15}$$

From the normal distribution, one can find that with $\alpha = 5\%$, $z_{\alpha/2} = z_{0.025}$ is 1.96, while with $\alpha = 10\%$, $z_{\alpha/2} = z_{0.05}$ is 1.645. As a rule of thumb, one can simply indicate that a confidence interval with a coverage of 95% is the $\hat{p} \pm 2{\cdot}$standard deviations.

When $p$ is close to 0 (e.g., when $p$ represents the error rate) or 1 (when $p$ indicates the accuracy rate), the coverage indicated by Eq. 4.14 is not always correct. Various textbooks specify that some additional constraints must be respected such as the product $s \cdot \hat{p}$ and $s \cdot (1 - \hat{p})$ must be larger than 5 or 10, or $s$, the sample size, must be larger than 50.

As shown by Brown et al. [43], this prescription is not fully correct, even for $p$ not so close to 0 or 1. As a result, the coverage of the confidence interval specified by Eq. 4.14 could be smaller than the exact value. For example, with $p = 0.2$ and $s = 25$, the coverage of the confidence interval (with $\alpha = 5\%$) is not 95% but only 88%. As another example, when $p = 0.5$ and $s = 40$, the achieved coverage is 91.9% instead of 95% or even 88.2% (with $p = 0.5$, and $s = 15$). Moreover, the exact coverage of the confidence interval presents an eccentric behavior that could be close to the specified value for some $p$ and $s$ values, and different for other values near to the previous ones.

A simple modification of the previous computation is proposed by Brown et al. [43]. First, instead of computing the proportion of success as given by Eq. 4.1, Eq. 4.16 must be applied (for a coverage of 95%).

$$Accuracy\ rate\ \hat{p} = \frac{Number\ of\ correct\ decisions + 2}{s + 4} \tag{4.16}$$

When the coverage of the confidence interval differs from 95%, another estimation of $\hat{p}$ must be employed as defined by Eq. 4.17. When $\alpha = 5\%$, one can find the previous formulation because $z_{0.025}$ is 1.96, or close to 2, and $1.96^2 \approx 4$. Thus Eq. 4.17 is approximatively equal to simplest form depicted in Eq. 4.16.

$$Accuracy\ rate\ \hat{p} = \frac{Number\ of\ correct\ decisions + \frac{z^2_{\alpha/2}}{2}}{s + z^2_{\alpha/2}} \tag{4.17}$$

For example, assume that the sample size $s = 25$ and that the system is able to classify 22 instances. The estimated accuracy rate according to Eq. 4.1 is $\hat{p} = 22/25 = 0.88$. To derive the 95% confidence interval, one can compute the standard deviation of $\hat{p}$ (Eq. 4.15), which is $0.88 \cdot (1 - 0.88)/5 = 0.02112$. The resulting confidence interval specified by Eq. 4.14 is $0.88 \pm 1.96 \cdot 0.02112 = [0.8386; 0.9214]$. As described previously, this interval might not have a coverage of 95%, but less than 95%. Estimating $\hat{p}$ by Eq. 4.16 gives $(22+2)/(25+4) = 0.8276$. The standard deviation of $\hat{p}$ (Eq. 4.15) is then $0.8276 \cdot (1 - 0.8276)/5 = 0.02854$. The resulting confidence interval with a coverage of 95% is therefore $0.8276 \pm 1.96 \cdot 0.02854 = [0.7716; 0.8835]$.

When comparing these two intervals, one can observe that the second is larger than the first one because it is built with a larger value for the standard deviation (0.02854 vs. 0.02112). Moreover, the estimated accuracy rate is lower compared with the second approach, 0.8276 vs. 0.88. The method described by Brown et al. [43] can be viewed as more conservative or less optimistic than the classical approach.

## 4.6 Statistical Assessment

After defining a performance measure and estimating its value and its associated confidence interval, it is important to infer whether or not two distinct classifiers (A vs. B) or an alternative of an existing classifier (A vs. A') achieve the same performance level.[2] To offer a precise answer, a formal decision rule must be defined and statistics will provide the required test to achieve this goal.

Let us start with the simple accuracy rate. When comparing two classifiers using this performance measure, the sign test [60] can provide a decision rule. This test is a specific variant of the binomial test, a more general test. As for other tests described in this book, the null hypothesis $H_0$ states that both classifiers result in the same performance level (no effect). The general idea under the sign test (or the binomial test) is the following. For each of the $s$ instances of problems to be solved,

[2]With all statistical tests, it is assumed that the data used to compute the effectiveness measures is randomly sampled from a given population. We postulate that it is always the case.

the sign test counts the number of times that the assignment resulting from both models differs. This number is denoted by $s'$. When the same attribution (correct or incorrect) is provided by both systems, this instance is not taken into account. Therefore, the value of $s'$ is smaller, at the limit equal to $s$. Moreover, $s^+$ represents the number of times that the first system proposes a correct assignment while the second system indicates an incorrect attribution. Similarly, with $s^-$ the number of decisions correctly assigned by the second classifier but incorrectly assigned by the first. Obviously, one can see that $s^+ + s^- = s'$.

Under the $H_0$ assumption (both schemes do produce the same performance), $s^+$ follows a binomial distribution with the parameters $p = 0.5$ (the probability of success) and $s'$ (the number of trials). When $H_0$ is true, one can expect having roughly the same number of $s^+$ and $s^-$. A large difference between these two numbers contradicts the null hypothesis and leads to reject $H_0$ (one classifier performs at a higher performance level than the other).

Consider the following example. From a sample of $s = 100$ attributions, $s' = 20$ decisions differ between the two classifiers. If $s^+ = 9$ (and $s^- = 11$), this value tends to confirm $H_0$, that both systems perform at the same level. Instead of computing the formal rule and looking at some statistical tables, the R software can compute the final result. As shown in Table 4.2, the sign test was applied when calling the function `binom.test()` and as arguments, the values of $s^+(9)$, $s'$ (20), together with the probability of success ($p = 0.5$).

The output of R summarizes the data and displays the $p$-value which is equal to 0.8. This is our main indicator to accept or reject the null hypothesis $H_0$. In this case, the $p$-value is high (0.8), indicating that the chance to observe a value as extreme as $s^+ = 9$ when $H_0$ is true is 80%. There is no reason to believe that $H_0$ is false. One can conclude that both systems really perform at the same performance level.

In a second example, $s^+ = 5$ with $s' = 20$ (bottom part of Table 4.2). In this case, the resulting $p$-value = 0.04 or 4%. This value corresponds to the chance of observing $s^+$ as extreme as 5 (assuming that both systems are working at the same performance level). The expression "as extreme as 5" corresponds to the case when $s^+$ = 0, 1, 2, 3, 4, 5 or 15, 16, 17, 18, 19, 20. The extreme cases are considered in both the lower and upper tail of the distribution (two-sided). All together, these 12 possible outcomes represent only 4% of the total distribution. This is possible but the probability is rather low (0.04 in the current case) and if one fixes the significance level at 5%, the null hypothesis is rejected. The conclusion is clear: both systems achieve different performance levels.

The second possible statistical test with the proposed performance measures is the $t$-test. In this case, two means are compared under the null hypothesis $H_0$ that they represent the same effectiveness or that both systems perform at the same performance level. The $t$-test is well-known [66] and different variants are possible. In our example, ten assignments have been achieved by two classifiers (System A and B), and the reciprocal rank (RR) has been selected as the performance measure. The higher the RR value, the better the performance. The RR value achieved by the two systems with respect to the ten instances are depicted in Table 4.3. Of course,

**Table 4.2** Two examples of the binomial test (or sign test) with R

```
> options(digits=4)
> binom.test(9, 20, 0.5)
   Exact binomial test
data: 9 and 20
number of successes=9, number of trials=20, p-value=0.8
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
0.2306 0.6847
sample estimates:
probability of success
          0.45

> binom.test(5, 20, 0.5)
   Exact binomial test
data: 5 and 20
number of successes=5, number of trials=20, p-value=0.04
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.08657 0.49105
sample estimates:
probability of success
           0.25
```

this is a very small example, and usually 50–100 instances are required to be able to clearly establish a possible performance difference between two classifiers.

In Table 4.3, the $t$-test function is called (`t.test()`) with the sample values A and B. The third argument specifies that we assume that the samples A and B have the same (or similar) variance (or standard deviation which is equal to the square root of the variance). If one thinks it is not the case, one can change the value from "True" to "False."[3] The last parameter (`alternative="two.sided"`) indicates that we do not have any theory or hint indicating that one classifier could perform better than the other. Thus, the alternative hypothesis, denoted $H_1$, can be stated as follows: "the two systems do not achieve the same performance level."

The output reported in Table 4.3 indicates that the $p$-value is rather low, 0.039 or 3.9%. With a very limited number of instances, this value indicates, under a significance level of 5%, that the null hypothesis must be rejected. The data does not support $H_0$, or more precisely, knowing that $H_0$ is true, the chance to observe the values in our samples (or more extreme ones) is around 3.9%. That could occur,

[3] Assuming different variances reduces the power of the $t$-test or its capability to quickly detect a performance difference. In the current case, the $p$-value will be slightly larger (4.7% instead of 3.9%)

but this event is rare and its occurrence probability is below 5% (the specified significance level). The conclusion is clear: the two classifiers do not achieve at the same performance level.

**Table 4.3** Example of a *t*-test with R

```
> options(digits=4)
> myData <- read.table("SystemAB.txt", header=T)
> myData
       A      B
 1    1.00    0.500
 2    0.50    0.250
 3    0.20    0.330
 4    0.50    0.250
 5    0.10    0.100
 6    0.20    0.160
 7    0.50    0.125
 8    1.00    0.200
 9    0.50    0.330
10    0.33    0.250
> attach(myData)
> mean(A)
[1] 0.483
> mean(B)
[1] 0.2495
> t.test(A, B, var.equal=TRUE, alternative="two.sided")
 Two Sample t-test
t = 2.2, df = 18, p-value = 0.039
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  0.01315 0.45385
sample estimates:
mean of x mean of y
   0.4830    0.2495
```

The resulting conclusion could be viewed as not very informative. In fact, the mean or the RR values achieved by System A were clearly higher (better) than System B (0.483 vs. 0.2495). Therefore, one can argue that the most pertinent test is not whether System A and B perform at the same performance level but to verify whether System A performs statistically better than System B. To achieve this, one must modify the last parameter of the function `t.test()` as shown in Table 4.4. In this case, the alternative is "greater" (or A > B, "A is greater than B"). Selecting this

option could be justified by the fact that a theory or other evidence indicates that the classifier A is working better and produces better assignments than System B.

The output of function `t.test()` in Table 4.4 indicates that the $p$-value is small (0.0195 or 1.95%). Therefore, the data (RR values for System A and B depicted in Table 4.3) does not support the null hypothesis $H_0$ (both systems perform at the same level) but tends to support clearly the alternative hypothesis ($H_1$: System A performs better than B).

**Table 4.4** Example of a unilateral $t$-test with R

```
> options(digits=6)
> t.test(A, B, var.equal=TRUE, alternative="greater")
   Two Sample t-test
data: A and B
t = 2.226, df = 18, p-value = 0.0195
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 0.05162689 Inf
sample estimates:
mean of x mean of y
 0.4830 0.2495
```

When applying the $t$-test, one must assume that the mean difference follows a normal distribution. This is not always the case, for example, with a small sample size. Empirical evidence, however, indicates that the resulting $p$-value is not really affected by the violation of this assumption even when considering other performance measures than the RR or different sample sizes [411]. Moreover, the $t$-test is more robust than other tests (such as the sign test, Wilcoxon signed rank test, or the bootstrap test). Thus we can recommend to use it.

## 4.7 Training and Test Sample

In practice, many stylometric models must first determine or learn some specific stylistic patterns able to discriminate between several authors or categories. For example, instead of considering all possible features, one must define a subset of the most pertinent ones. This constraint implies that some textual data must be reserved for this learning stage. This subset is called the *training set*. It is unfair (or more precisely biased) to reuse this training set to evaluate the effectiveness of the system. One cannot simply employ all the instances to learn the underlying parameters of the classifier and then reuse the same set of instances to evaluate its performance.

Therefore, two disjoint subsets of the original dataset must be created: one for the learning stage (the *training set*) and the other to evaluate the classifier called the *test*

*set* or *validation set* [179]. Such an evaluation strategy is called *hold-out*. Using the same instances to train and test will produce a biased and too optimistic performance called *resubstituting performance estimate*. This biased estimation appears more often than supposed, in part because in some cases the size of the available data is rather small.

Thus, splitting the original dataset into two disjoint parts implies that less data is available for the training as well as for the evaluation (the test set). However, one can be lucky and have a large sample of instances in the original dataset. In such cases, one can save, for example, one-third for the test set, and two-third for the training. Usually, more information is provided to the training set to ensure that the system can learn something from the data.

When the dataset size is limited, it is however possible to have a large number of instances in the training set together with all instances appearing in the test set. This evaluation approach is called *k-fold cross-validation* [179, 420]. In this case, the original dataset is subdivided into $k$ roughly equal-sized subsets. The intersection between all of them is nil. One after the other, each of these $k$-folds represents the test set, the remaining folders constitute the training set. This procedure generates $k$ performance measures, one for each $k$-fold. The overall effectiveness is obtained by computing their arithmetic mean. As possible values for $k$, practical and theoretical considerations indicate that $k = 10$ or $k = 5$ are reasonable choices [156].

Let us assume that the sample size is 100 and $k = 10$. After a random shuffling of these 100 instances, the first fold is composed of the 1st to the 10th instances. Thus, the classifier is built with instances from rank 11th to 100th, and the test set contains examples from the 1st to the 10th position. The second fold is composed of instances at rank 11th to 20th. The training set contains examples from the 1st to the 10th together with instances at position 21th to 100th. The second fold is then used to test the new classifier. And this iterative process continues to the tenth fold.

A last but important remark. The dataset applied to train and test a classifier must closely reflect the application context. For example, to identify the true author of Ferrante's novel (see Chap. 8), the dataset must correspond to novels written in Italian for adult readers and published in the period 1990–2018, corresponding to the publication period of Ferrante's novels. Knowing that a novel contains, in mean, more than 10,000 words and that the spelling is verified, the data quality and length will not invalidate the methods. As candidate authors, one needs to select all possible known Italian writers depicting some relationship with Ferrante's novels (e.g., being a woman, lived in Naples or have known this city, etc.).

In forensic cases, the situation is rather different and harder. The text for which one must determine the author's demographics or even name is usually short and the orthography could present several errors [289, 291]. Those errors, or pattern of errors, could also provide a hint about the true author. Nonetheless, obtaining many texts from the candidate authors is problematic. Therefore, all possible stylistic features are extracted from the available document. Moreover, the text genre available to train the classifier could be different from the text genre present in the criminal record (e.g., a threatening e-mail, a suicide letter). Thus cross-text genre investigation must be conducted, leading to a higher imprecision level [54].

## 4.8 Classical Problems

After describing the evaluation methodology, some well-known authorship attribution problems, mainly related to the literature domain, could form an interesting testbed. In addition, the CLEF PAN campaigns have also produced some useful test collections to evaluate different stylometric problems such as determining the author's gender, age range, or some of his psychological traits [40, 280]. These test corpora will be presented in the next section.

Our first example is the *Federalist Papers* corpus described in Sect. 2.2. Various studies have been published to resolve this case, in particular the seminal book of Mosteller and Wallace [273] which was the first to apply statistical approaches, Bayes models in this case, to stylometry problems. As other works on this topic, one can mention Tweedie et al. [410] who suggest applying a neural network using only 11 words while Fung [124] proposes considering only three word-types (*as*, *our*, and *upon*). As the *Federalist* corpus represents a good benchmark, it was used to evaluate the effectiveness of different models as explained in [162, 183, 268, 327, 338].

As a second example, the *Pauline Epistles* or *Letters of Paul* correspond to 14 letters attributed to Paul, the Apostle. This corpus was written in Ancient Greek between around AD 47 (estimated year for the *Epistle to the Galatians*) to around AD 68 (*Second Epistle to Timothy* or *2 Timothy*). Belonging to the New Testament, these letters are the oldest Christian writings. However, the name *Paul* appears only on 13 texts. The last one (*Hebrews*), written anonymously, corresponds to Paul's doctrine, but it is generally admitted that the true author is not Paul (pseudepigrapha or false attribution to Paul). Biblical scholars have however debated the authorship of these letters and they disagree about the number of letters that can really be attributed to Paul. One of the difficulties of this question is the text length with the longest (*Romans*) containing 8233 words while the shortest (*Philemon*) is the most problematic to attribute with only 388 words. Such a short length represents a real challenge. Some experts agree that only four are genuine letters of Paul (namely *Romans*, *1 Corinthians*, *2 Corinthians*, *Galatians*). On this set, a majority of scholars have added *Philippians*, *1 Thessalonians*, and *Philemon* as genuine texts of Paul. This latter group forms a set of seven letters. The remaining six letters (without *Hebrews*) are subjects of numerous discussions. In a stylometric perspective, this question could be viewed as a verification problem [214, 349]. The problem formulation is the following: Assuming that four letters have been written by Paul, are the remaining ones written by the same author?

One can also mention the authorship related to *The Book of Mormon*, the sacred text of the Latter-Day Saint movement, first published in 1830 by the Mormon prophet Joseph Smith. But is J. Smith the true author? The Mormon tradition specifies that J. Smith dictated his thoughts to scribes and one can view this book as a verbatim transcript of Smith's words. We could also assume that the scribes were not simple copyists but must be viewed as collaborative authors. The issue is however more complex knowing that we do not have enough reliable texts authored, without any doubt, by J. Smith. Thus one research question is to determine if

the *Book of Mormon* was written by a single author (J. Smith) versus a multiple authorship [161]. Jockers [181] analyzes the *Personal Writings of Joseph Smith*, a collection of around 100 letters, diaries, histories, and other documents containing between 112 and 2300 words. He found that these personal writings are not all genuine samples of Smith's style but certainly reflect Smith's doctrine.

One of the most famous authorship questions is related to Shakespeare (1564–1616) [65, 104, 239, 390]. In France, some well-known plays written by Molière (1622–1673) also raise an authorship problem [227]. Some similarities can be found between these two disputes. First, the legal notions associated with the copyright differ greatly compared to the contemporary concept. Compared to our legal framework, the term *author* covered a different reality at that time. Moreover, producing a joint work was not a rare event. For example, Molière and P. Corneille produced a collaborative play *Psyché* in 1671. This collaborative practice was however more frequent with Shakespeare. As examples of plays having two authors, one can mention *Henry VIII* (Shakespeare and Fletcher), *The Two Noble Kinsmen* (Shakespeare and Fletcher), *Timon of Athens* (Shakespeare and Middleton), *Titus Andronicus* (Shakespeare and Peele), and *Pericles* (Shakespeare and Wilkins).

Second, both authors have been put on a pedestal, mainly during the nineteenth century. Shakespeare could be viewed as one of the greatest writers of all time, and the French language is called Molière's language (*la langue de Molière*). The same is also true for English being called Shakespeare's language.

Third, many external evidences have been put forward to confirm or deny authorship of some of their plays. For example, at their deaths, one cannot find writings, unfinished plays, or even a real library for such renowned writers. Moreover, their biographies are not fully clear and some passages of their lives still raise uncertainty. For example, as France and Italy appear in many places in Shakespeare's works, can we find historical evidence that the Bard of Avon had visited France or Italy? Moreover, could his social networks and education explain his familiarity on law, politics, astronomy or even the rules of the king's court needed to write some passages in his plays [239]?

Fourth, in both cases a passionate debate discusses this question, often based on authoritative arguments, a single document or new discovered evidence. According to [391], Shakespeare's authorship debate draws a large audience with more than 4000 books and articles published on this topic.

Let us take a look at some examples of those external arguments. Shakespeare died in 1616, and among his last plays, one can mention *Macbeth* (written in 1606 according to [392]), *Cymbeline* (1610), *The Tempest* (1611), or *Henry VIII* (1613). When supporting that the real author behind Shakespeare's works is either C. Marlowe (1564–1593) or E. de Vere (17th Earl of Oxford, 1550–1604, the favorite name), it is problematic to explain how some plays could appear after the death of their supposed author. It seems clear that both Marlowe and de Vere cannot be good candidates anymore. But various arguments have been advanced to justify this time gap. First, one can specify that these works (or some of them) cannot appear in Shakespeare's canon because they have been written by another unknown writer. Second, one can put forward that the writing dates indicated previously are

simply wrong and all these works have been written before 1604 (or even before 1583). As a third reason, one can say that either Marlowe or de Vere had left a large amount of drafts of those plays and therefore the final version was released after their death. Fourth, the supposed death in 1583 (or 1604) was in fact a fabrication, and Marlowe (or de Vere) was able to continue to write from a hidden place [104]. All those external argumentations are not related to stylometric models and are usually difficult to either confirm or deny.

Finally, a careful look reveals some differences between Shakespeare's and Corneille's cases. With Shakespeare, numerous authorship candidates (more than 50)[4] or even a group of writers have been suggested [390]. For Molière, only one name appears P. Corneille[5] (1606–1684). Moreover, with the Bard, the humanities scholars scrutinize which parts of several plays correspond to a genuine Shakespeare style or to a possible or known second writer. This is not the case with Molière's work. Except for *Psyché* (1671), we admit that all the disputed plays have been written by a single author.

Edgar Allan Poe (1809–1849), a well-known American writer who succumbed to an abrupt death, also presents some authorship questions. Various manuscripts have been found after his death and appeared as "possible Poe." In addition, this authorship issue is rendered more complex by the occurrence of various pseudonyms (e.g., Quarles, a Bostonian, Edgard A. Perry). In a recent study, Schöberlein [352] analyzes the authorship of thirty-two prose texts and ten poems. As a related question, Schöberlein studies the style of the Paulding-Drayton review defending Southern slavery in the United States. Usually, this article is attributed to N.B. Tucker, but Poe could have been involved in the writing. Applying the rolling Delta [333] (see Sect. 7.5) to this work, one can find Poe's style in the first part of this article, confirming a collaboration. Or, as indicated by Schöberlein "Poe went beyond the usual tasks of an editor and apparently reworked Tucker's words into a final product."

In more recent years, some authorship problems still appear in the literature domain and three examples could be mentioned. First, Joanne K. Rowling published a novel under the penname Robert Galbraith entitled *The Cuckoo's Calling* (April 4th, 2013). The author of the *Harry Porter* series wanted to demonstrate that she was able to script on a different topic and text genre and adopt a distinct style. She decided to write a crime fiction. But the mystery did not live long. On July 13th, the *Sunday Times* revealed that the real author was J. K. Rowling [191, 193], a fact confirmed by the author herself.

A second instance occurred in France where Romain Gary was a well-known writer. But the critics saw him as a "has-been writer" with "a boring style." To demonstrate that his literary career was not finished, Gary decided to craft a novel

[4]The last proposition appears in *The Atlantic* in June 2019. E. Winkler suggests that an Italian woman called Emilia Bassano was the true author.

[5]In French literature, there are two authors named Corneille, namely the brothers Pierre (1606–1684) and Thomas (1625–1709).

under the pseudonym Emile Ajar (*Gros câlin*, 1974). With the next novel published under the same *nom de plume* (*La vie devant soi*, 1975), he won the Congourt[6] award. Just before dying (1980), Gary disclosed that he was the author behind Ajar's novels. As for the Rowling's case, this attribution or verification problem could be investigated by comparing several novels [226, 295].

The background of the third case is Italy, more precisely the city of Naples, where the story of *L'amica geniale* (2011) (*My Brilliant Friend*, 2012) is located, written by Elena Ferrante. This novel was the first one of a tetralogy that garnered worldwide success. The essential question is to know who is the real author of these novels. Chapter 8 is devoted to this problem (see also [407]).

## 4.9 CLEF PAN Test Collections

The real examples presented in the previous section correspond to true and interesting cases to enhance our knowledge, but they are not fully pertinent to evaluate a stylometric model and to compare its effectiveness with others. Scientific research requires two additional conditions. First, the experiment must be repeatable, usually by, at least, one other researcher. Each new discovery must be confirmed independently by other sources. In our previous examples, the underlying novels and texts are not always freely available (partially due to copyright). Second, the exact same data must be used by the different researchers. In addition, some preprocessing choices are not always clearly described or not justified theoretically. To fulfill theses gaps, the CLEF initiative (www.clef-initiative.eu) [118] was launched in 2000 to promote international evaluation campaigns in order to encourage research in natural language processing, to share datasets and knowledge and to support commercial applications.

A typical CLEF campaign is organized as follows. Each year a set of seven to nine tracks are selected by a scientific steering committee. Each track is focused on a subset of related tasks. For example, the CLEF PAN track is concerned with stylometric models and applications. For each track, the data is always freely available and the texts are written in different European languages. Even if English is the most frequent one, one can find datasets written in Spanish, Italian, German, French, Dutch, or even in Arabic.

Each year, the track coordinators propose a set of tasks to be pursued, usually between two and four. For example, in the CLEF PAN track, the first task could be an authorship attribution, the second a plagiarism detection, and the third might target an author profiling problem. Associated with each task description, one (or a few) performance measure and one or more datasets are provided. This data collection is divided into two parts. The training set is available around February to allow

[6]This prestigious French award can only be won once, but Gary obtained it twice, in 1956 under his name, and in 1975 under the penname E. Ajar.

participants to elaborate and train their models. In May, the test set, without the correct answers, is made available. Then each participant has around 1 month to send their results to the track coordinators. The correct answers are released around the end of June. During the conference (held in September in Europe), an overview of each task is provided by the track organizers and each participant can present their own solution. Based on these publications, one can have a good idea of the effectiveness of different possible models to solve a task, with their advantages and drawbacks. After the conference, the datasets (both the training and test sets) are freely available on the Internet for further investigation.

For the different CLEF PAN tracks (more information available at `www.pan.webis.de`), a good overview of the different tasks proposed over the last 10 years can be found in [311]. Over the years, the followings tasks have been proposed: authorship attribution (2011–2012, 2018–2019), authorship verification (2013–2015), author's age prediction (2013–2016), author's gender identification (2013–2019), personality prediction (2015), style change detection (2017–2019), author clustering (2016–2017), and language variety analysis (2017). In order to have an overview of the datasets, Table 4.5 reports those related to the author profiling tasks over the years.

**Table 4.5** Example of test collections (gender prediction) extracted from the CLEF PAN campaigns

| Year | Text Genre | Instances Train/Test | Training Set Male/Female | Test Set Male/Female | Mean Length |
|---|---|---|---|---|---|
| 2013 | Blogs | 13,106/25,440 | 11,524/11,291 | 24,009/23,918 | 871 |
| 2014 | Blogs | 147/78 | 1257/1021 | 527/590 | 2788 |
| 2014 | Tweets | 306/154 | 119,611/81,821 | 46,754/46,922 | 12,866 |
| 2015 | Tweets | 152/142 | 6986/7180 | 6268/6910 | 1425 |
| 2016 | Tweets | 434/78 | 134,961/105,720 | 527/590 | 14,816 |
| 2017 | Tweets | 3600/2400 | 180,000/180,000 | 120,000/ 120,000 | 1772 |
| 2018 | Tweets | 3000/1900 | 150,000/150,000 | 95,000/95,000 | 1781 |
| 2019 | Tweets | 2060/1320 | 103,000/103,000 | 66,000/66,000 | 1590 |

In Table 4.5, the first two columns indicate the year and text genre of the corresponding test collection. Under the label "Instances," the value indicates the number of documents (or problem instance) for which the system must determine the author's gender. Each instance corresponds to a set of tweets or blog posts appearing, respectively, in the training and test collection. The evaluation is not performed on a per tweet basis, but each document contains between 40 and 100 tweets. The next two columns indicate the number of tweets/blog posts in the training and test set as specified in the evaluation campaigns. As each instance is composed of between 40 and 100 tweets, the numbers depicted in the last two columns are greater than the number of instances. The last column specifies the mean number of tokens per document (e.g., set of tweets). The mean length appearing in some corpora is rather small, rendering the correct attribution harder.

## 4.10 Evaluation Examples

Using the performance measure and evaluation methodology explained previously, one can assess the three authorship models described in Chap. 3. The problem is to determine the true author of the 12 disputed articles appearing in the *Federalist Papers*. As answer, each classifier will provide a ranked list of author names, and occurring in the first position, one can find the most probable writer. As the number of candidates is limited to three, the accuracy rate has been selected as evaluation measure. Thus, when Madison appears in the first position, the answer is tagged as *correct*, and *wrong* for the other names.

The effectiveness achieved by these authorship attribution classifiers is reported in Table 4.6. For each model, some parameters must be defined, usually inside a range of possible values. First with the Delta model, one must stipulate the number of the most frequent word-types (MFWs) used as stylistic markers. In this list, punctuation symbols are considered as tokens. As this limit is not clearly fixed in the articles, Table 4.6 depicts values from 50 to 500 MFWs. To define these feature sets, only the 70 training articles[7] have been taken into account. Limited to 50 or 100 MFWs, two incorrect assignments have been produced, both incorrectly attributed to Hamilton (Paper #55 and #56). With 150–300 MFWs, a single wrong attribution can be observed (either Paper #55 or #56). With 400 or 500 MFWs, all the answers were correct.

This first set of experiments were conducted without any sophisticated feature selection procedure. The simple criterion was the occurrence frequency, which is usually effective for authorship attribution. Two additional remarks are worth considering. To be effective, the number of terms must usually be between 200 and 500 [339]. Second, even if such a feature set is called functional words, many entries are not function words. For example, in the set of the 50 MFWs extracted from the *Federalist Papers*, one can find terms related to the topics such as *states*, *government*, *power*, or *state* (see Appendix A.1).

As an additional experiment, the Delta model was applied with 34 word-types suggested by the long list of Mosteller and Wallace [272]. Those word-types have been more carefully selected according to their power to discriminate between Hamilton's and Madison's styles. With this reduced feature set, Delta generates a single error (Paper #55 assigned to Hamilton). Here too, the words selected are not all functional words, for example, *consequently*, *vigor*, *work*, or *language* [272]. As a last experiment, the list of functional words suggested by Antonia et al. [11] and composed of 192 entries was used. In this case, all instances are correctly classified.

To verify whether the performance differences can be viewed as significant, the sign test was applied. Based on it, one cannot find any statistically significant performance differences between these six Delta implementations. In fact, the

[7] Namely 51 for Hamilton, 14 for Madison, and 5 for Jay.

**Table 4.6** Evaluation of the *Federalist Papers* (12 disputed articles)

| Method | Features | Incorrect (Hamilton) | Incorrect (Jay) |
|---|---|---|---|
| Delta | 50 | #55, #56 | |
| Delta | 100 | #55, #56 | |
| Delta | 150 | #55 | |
| Delta | 200 | #56 | |
| Delta | 300 | #55 | |
| Delta | 400 | | |
| Delta | 500 | | |
| Delta | Mosteller and Wallace [272] | #55 | |
| Delta | Antonia et al. [11] | | |
| Labbé | instance | #49, #52, #56, #58 | |
| Labbé | profile | | |
| KLD | Zhao and Zobel ($\lambda = 0.1$) | | |
| KLD | Zhao and Zobel ($\lambda = 1$) | | #50 |
| KLD | 200 ($\lambda = 0.1$) | | |
| KLD | 200 ($\lambda = 1$) | | |
| KLD | Antonia et al. ($\lambda = 0.1$) | | |
| KLD | Antonia et al. ($\lambda = 1$) | | |

number of instances in the test set is too limited (12) to obtain, from a statistical point of view, a significant difference.

As a second authorship model, the Labbé's intertextual distance was evaluated in Table 4.6. This model specifies that the text length must be, at least, 5000 tokens to be performed correctly. This is not the case with these newspaper articles (mean length: 2480 tokens). Considering the distance between individual articles (instance-based evaluation), this classifier generates four incorrect assignments, all attributed to Hamilton.

As a variant, all articles of each author have been concatenated to generate a single author's profile. The style of each author is then represented by a large number of tokens, more than 10,000 for both Hamilton and Madison, just slightly less than 10,000 for Jay's profile (precisely, 9393). In a second step, for each disputed article, the Labbé's intertextual distance is computed to define the closest profile. Adopting this representation, Labbé's classifier does not produce any errors. The performance difference between these two implementations can be viewed as statistically significant (sign test, $p$-value = 0.0125).

The KLD approach suggested by Zhao and Zobel [431] is our third authorship model. As feature set, Zhao and Zobel's list (344 word-types), the 200 MFWs or Antonia et al.'s list (192 function words) has been employed. To estimate the occurrence probability, the Lidstone's approach was applied with $\lambda = 0.1$ (as suggested in Chap. 3) or with $\lambda = 1$, a value corresponding to Laplace's smoothing. Using Zhao and Zobel's list, the classifier produces one error when $\lambda = 1$ (Paper #50 attributed to Jay), but no errors when selecting $\lambda = 0.1$ (as suggested previously).

With the 200 MFWs or Antonia et al.'s list, this strategy produces no error. From a statistical point of view (sign test), one cannot detect any performance difference between the two solutions based on Zhao and Zobel's list. The reduced number of instances in the test set (12) explains the difficulty in detecting a statistically significant performance difference when the number of errors is small.

The effectiveness is of prime importance, but the proposed attribution must also be explained to the final user. To determine the reasons justifying an attribution to a given author, the contribution of the different terms in the distance between the disputed text and the different author profiles must be examined. As an example, Table 4.7 displays the contribution of the six words having the highest Z score values in Madison's (on the left) or Hamilton's profile (on the right). These values have been obtained by considering the 50 MFWs. In the two parts of this table, clearly the word (first and fifth columns) together with the Z score values obtained within Paper #54 surrogate and under Madison's or Hamilton's profile. Under the label "Delta," the difference between the two Z score values is provided. This value also indicates the increment of the distance between Paper #54 and the corresponding author's profile.

**Table 4.7** Delta method applied with Paper #54

| Word | Paper#54 | Madison | Delta | Word | Paper#54 | Hamilton | Delta |
|---|---|---|---|---|---|---|---|
| on | 1.926 | 1.331 | 0.595 | this | 1.217 | 0.344 | 0.873 |
| by | 1.257 | 1.029 | 0.228 | in | 1.528 | 0.334 | 1.193 |
| ; | -0.511 | 0.762 | 1.273 | of | -0.843 | 0.297 | 1.140 |
| no | 0.833 | 0.586 | 0.247 | to | -1.528 | 0.249 | 1.778 |
| the | 0.355 | 0.571 | 0.216 | a | -0.810 | 0.245 | 1.056 |
| states | 2.104 | 0.542 | 1.562 | an | -0.468 | 0.237 | 0.705 |

In the first row, the word *on* obtains a Z score of 1.926 in Paper #54. This term also achieves the highest Z score under Madison's profile with a value of 1.331, and the increment of this word with Madison's profile is 0.595. The largest contradiction to assign this paper to Madison is supported by the semi-colon (;) having a negative Z score in Paper #54 (−0.511, signaling that it appears less frequently than the mean) and with a positive value under Madison's profile (0.762).

When inspecting the six words having the largest Z score values with Hamilton's profile, the Delta values (last column) are all relatively large, signaling some contradiction between the usage in Paper #54 and those in Hamilton's profile. For example, the word *to* has a negative Z score value in Paper #54 (−1.528 and thus it occurs less frequently than the mean) compared to Hamilton's profile (0.249).

With the data depicted in Table 4.7, one can see that even with the true author some differences appear between the text representation and the author's profile. Under the fourth column labeled "Delta," some words present a relatively large increment (e.g., with the semi-colon or "states") when compared to the true author's profile (Madison).

In a scientific perspective, an experiment suggesting an attribution should be repeated or verified. Therefore, the data used in an experiment must be freely available, when possible. In a US court, to admit an authorship attribution model as an admissible testimony, a strict protocol must be specified and followed. In addition, the error rate must be estimated and validated through empirically established procedures [54, 64]. To fulfill (even partially) these desiderata, one must repeat our previous experiments with other test collections written in different text genres and time periods, with varying text lengths and a larger number of authors [54].

Finally, one must not over-generalize the result of a single or a few experiments. A given attribution is a conjunction of a feature set and a classifier. Both are essential to produce the final assignment. In addition, the unknown characteristics of a corpus might favor one attribution scheme over the others. Therefore, a set of additional experiments is required to verify the sensibility of the proposed attribution model to different parameters (e.g., text genre or length, language, preprocessing, feature sets, etc.).

# Chapter 5
# Features Identification and Selection


Each classification system is based on two main components, namely a set of features and a classification model. This chapter is dealing with the first component. In the previous chapters, we implicitly admit that stylistic markers are simply isolated words. Such word-based modality can be enriched by adding features extracted from the character-based, syllabic, topical, syntactic, and layout modality. In this view, Sect. 5.1 lists and comments on possible word-based stylistic features, namely isolated words, $n$-grams of words, dedicated wordlists, POS tags, or sequences of them. As explained in Sect. 5.2, more recent studies have proposed a larger spectrum of strategies to identify specific stylistic features depending on the target applications. This section exposes the use of $n$-grams of letters, text distortion, or DNA-based representation techniques.

Enumerating all possible features does not mean that all of them are pertinent and effectively reflect the stylistic differences. By reducing the number of attributes, one can simplify the analysis of the proposed assignment, speed up the computation, and usually improve the overall performance. Five main paradigms of feature selection procedures have been proposed. First, one can derive a pertinent attribute subset by considering only their intrinsic properties. In this perspective, Sect. 5.3 exposes frequency-based metrics to reduce the number of features. Second, as presented in Sect. 5.4, filter-based approaches take account of the capability of a feature to discriminate between categories. For some authors, these first two families could be regrouped together. As a third variable selection paradigm, Sect. 5.5 proposes an overview of wrapper approaches. In this case, the pertinence of a feature subset is analyzed by considering its impact on the classification performance. Therefore, instead of selecting the attributes before applying a classifier, its effectiveness according to the underlying learning scheme is computed to derive the best feature subset. Fourth, the features can be selected during the learning stage (embedded feature selection) as proposed by some machine learning strategies and described in the next chapter. As the last paradigm, one can consider different ad hoc methods or manual approaches. For example, a linguistics theory can ascertain the set of useful



J. Savoy, *Machine Learning Methods for Stylometry*,
https://doi.org/10.1007/978-3-030-53360-1_5

features. In addition, after computing the correlation coefficients between pairs of attributes, one can remove redundant ones.

One must be careful when specifying that the PCA approach (see Sect. 3.5) is a kind of feature reduction approach because the generated space is limited to a few principal components. Usually, the figure displaying the different profiles using the first two principal components could lead to the idea that only two features are enough to discriminate between the different stylistic categories. One must remember that each principal component is a linear combination of a large number of the original features. Some of them could have a null coefficient and thus they are ignored by the corresponding principal component.

Finally, Sect. 5.6 presents two strategies that can be applied to characterize the vocabulary of an author (or category) as well as the *lexis* related to a disputed text. Such procedures can reveal the similarity between samples of texts and could be useful to explain a proposed attribution.

## 5.1 Word-Based Stylistic Features

To extract a broad set of stylistic features, one can choose different linguistic modalities or categories of linguistic items. As indicated in Chap. 3, the first source of stylistic attributes corresponds to the words and expressions an author likes or dislikes. As mentioned previously, the term *word* could correspond to the form present in the text or to the lemma (or headword). As English has simple inflectional rules, the difference between the two forms is usually related to the suffixes "-s," "-ed," and "-ing" (without considering exceptions such as *talk* and *told* or *child* and *children*). For other languages, the differences between the surface form and the lemma could be more complex, for example, with languages having grammatical cases indicated with a suffix (e.g., in Latin with *rosam*, *rosae*, *roasas*, or *rosarum* vs. *rosa*). The identification of the lemma from a given surface form can be obtained by a morphological analyzer or more frequently by a POS tagger as described below.

Besides the surface form or the lemma, one can opt for representing each word by its stem. Such a result is performed by a stemming procedure well-known in information retrieval. This procedure is usually applied without considering the context of the word leading to a fast execution. The general idea is to look for some predefined patterns at the word ending and to remove them by respecting some constraints and rewriting rules. For example, from *running*, one can remove the final "-ing" because the resulting string is larger than three letters (e.g., which is not the case with *king* or *ring*). After this truncation, if the last two letters are the same, remove one (thus instead of *runn*, *run* is returned).

For the English language, one can apply the light Harman's stemmer [150] to remove the plural suffix for nouns (or the final "-s"). For some applications, also removing derivational suffixes could be an advantage (e.g., the stem *power* could be extracted from *powerful*, *powerfully*, or *powerless*). In English, the Porter's stemming procedure [305] represents a frequently proposed solution. However, there is

no guarantee that the resulting stem is the correct one (e.g., from *organization* the resulting stem is *organ*) [112]. For other languages, one can find such a general stemming procedure on the Internet.[1]

Isolated words constitute a frequently applied approach in various stylistic applications. But the style is not limited to the word choice but also in their combinations. Thus, as an attribute, one can consider sequences of two adjacent words denoted bigrams. Longer word sequences could be considered, for example, trigrams (three words) or, more generally, $n$-grams of words.

Various implementations could generate different representations based on word $n$-grams. First, usually the punctuation symbols are not taken into account. Second, the strong punctuation denoting the end of a sentence (., ?, !) stops the $n$-gram generation. Third, instead of considering adjacent words, the $n$-gram sequences could be created according to the presence of some word categories or according to a wordlist (e.g., the 100 MFWs). For example, from the sentence "The fox of the forest jumps over the river." the following bigrams can be generated: "the fox," "fox of," "of the," "the forest," ... "over the," "the river". Another implementation would add the bigram "river." to include the punctuation.

When identifying the style with word $n$-grams, one can detect sequences appearing very often in one category and rarely in the others. For example, the sequence "said the police" appears frequently in newspaper articles, while the trigram "God bless America" could be viewed as evidence in favor of a US presidential speech (but since R. Reagan). When considering the most frequent word bigrams or trigrams, one can discover some syntactic patterns, for example, "of the," "that I," or "a lot of." Representing an authorial style with a vector of such items can be effective [28]. Moreover, offering the short context of a word provides hints about its usage. As the meaning of a single word could be ambiguous, a $n$-gram could reveal the usage and author's intent. For example, what is the meaning of "drop"? As a single word, the meaning is not clear, but look at the following short context: "he drops a goal," "a drop of whisky," "a steep drop," "the car makes the drop," "the pen drops from the table," "she drops Bob," or simply "drop dead" [72]. As a real example, one can mention the use of the word *nuclear* during the US electoral campaign in 2008. J. McCain employs this noun frequently. What was the underlying purpose? Nuclear weapons? Energy? DNA? [335]. The ambiguity is removed when trigrams are used to represent the speeches. The frequently used sequence *nuclear power plant* indicates the proposed solution to solve the energy issue, a topic closely related to the author, J. McCain. Considering topical words as an additional source of evidence about an author's style was suggested by different studies [296, 306, 337].

A text representation could be based not on a fixed length word $n$-gram but take account jointly of uni-, bi-, and trigrams leading to *variable length* word $n$-grams. In this perspective, Antonia et al. [11] demonstrated even if isolated words tend

[1]For example, see at http://www.members.unine.ch/jacques.savoy/clef/ or http://www.snowballstem.org/.

to produce the best performance, word bigrams or trigrams could be useful when considering more rare stylistic markers.

Sequences of word $n$-grams are not synonyms of word collocations. The latter is defined as a sequence of words that often co-occur together. Many of them can be detected by considering short sequences of adjacent words such as *crude oil*, *olive oil*, or *heavy traffic*. Other collocations can have one or more words between their components such as with "bread and butter" or "bank and money" [95, 364].

As a drawback, the large number of possible attributes raises some concerns. When, from a given corpus, one can find $|V|$ word-types (let us say 5000), one can find $|V|^2$ bigrams (25,000) as well as $|V|^3$ trigrams (125,000,000). Of course, many of them will simply never appear (e.g., the trigram "of the of") or occur with a very low frequency (see Zipf's law in Sect. 2.3). But considering only 50% of them, the number of possible features is still huge, rendering an interpretation of the assignment more difficult to infer and explain.

As terms[2] or stylistic items, one can also consider the POS tags or sequences of such tags. To achieve this, a shallow[3] POS tagger should attribute the appropriate tag to each token present in a sentence. The most frequently used set of tags are those described in [252] and associated with the Penn Treebank project or with the Brown corpus [123]. For example, from the sentence "Our/PRP$ energy/NN policy/NN is/VBZ creating/VBG new/JJ jobs/NNS ./.". Tags may be attached to nouns (NN, noun, singular, NNS noun, plural), verbs (VB, base form, VBG gerund or present participle, VBZ 3rd-person singular present), adjectives (JJ), personal pronouns (PRP), prepositions (IN), determiners (DT), and adverbs (RB). To these real POS tags, one can also consider symbols (SYM), foreign words (FW) and add a specific mark used to denote the beginning and the end of the sentence. Habitually the same symbol is employed for both functions.

With this information, the lemma can be derived by removing the noun plural suffix (e.g., "jobs/NNS" → "job/NN") or by substituting inflectional suffixes of verbs (e.g., "creating/VBG" → "create/VB"). Evidently, the number of tags could be limited to nine main POS categories (namely, noun, verb, adjective, adverb, pronoun, determiner, preposition, conjunction, and interjection). One can also include the morphological information such as the number for nouns (singular or plural), also the tense and person for verbs (e.g., 3rd person singular present). This solution was adopted in our previous example. In addition, one can include names, foreign nouns, numbers, dates, or symbols ($, €, ¶, ©).

Considering the word together with its POS tag is useful to distinguish between different usages of the same word and thus to remove some lexical ambiguities. For example, *to* could be a preposition (e.g., to Boston) or used with the infinitive form

[2]Term is used in a general sense to indicate a stylistic feature such as a word, a $n$-gram of words, a sequence of POS tags, a $n$-gram of characters, etc.

[3]A shallow parser identifies the constituent parts of a sentence (nouns, adjectives, verbs, etc.) without generating the full parse tree (or syntax tree) of the sentence (indicating the subject, verb, and objects or complements).

of a verb (e.g., to see). The ambiguity is also present with other frequently used words such as *that* (as a determiner in "that book" or as a conjunction in "I know that he is in Paris" or even as adverb "I can't wait that long"). As suggested by Burrows [46], the distinction between the different POS associated with frequently occurring words can discriminate between different authorial voices as shown in [16]. For example, one author could favor the use of *that* as a determiner, while the second as a conjunction even if in both cases the relative frequency is similar.

Overall, the number of distinct tags is rather limited (e.g., 30–50) compared to the number of word-types. Even when considering $n$-grams of POS tags (e.g., IN_DT, JJ_NN_NN), the number of stylistic markers that can be generated is reasonable. For example, starting with 40 distinct tags, one can obtain a maximum of $40^2 = 1600$ bigrams or $40^3 = 64{,}000$ trigrams (and many of them will never appear, such as "determiner–preposition–determiner").

In a study based on literature text segments written in English or French, Kocher and Savoy [211] found that isolated POS tags do not perform well compared to word unigrams (see also [146]). Taking account of bigrams or trigrams of POS tags could acceptably characterize the underlying stylistic aspects of texts. The achieved performance is then close to those obtained with isolated words.

## 5.2 Other Stylistic Feature Extraction Strategies

Obviously, words constitute a reliable source of stylistic markers. However, one can view them as molecules that can be decomposed into smaller items. Thus, as stylistic items, many studies suggest considering the letters, starting with isolated ones, but sequences of them tend to provide more effective stylistic markers. Usually the distinction between uppercase and lowercase letters is ignored and punctuation marks and symbols are removed. However, instead of ignoring them, one can regroup all punctuation symbols under the same sign or according to a few classes (e.g., three with the first representing periods, the second for commas, and the third for all remaining punctuation symbols) [296].

When generating letter $n$-grams and working with languages presenting diacritics (e.g., é, ü, ì), it is not always clear whether or not the accents must be removed. Finally, the separation between words is indicated with a special character (e.g., the space or the underscore) to clearly indicate the word boundaries. This special symbol is useful to detect prefixes or suffixes. Let us take an example to illustrate this feature generation. From the sentence "To be or not to be," the following letter trigrams can be generated: "_to", "to_", "o_b", "_be", ..., "o_b", "_be", "be_".[4] Instead of representing a text with fixed letter $n$-grams, variable length letter $n$-grams are applied (usually with $n$ varying from 2 to 6).

[4] In the Appendix, Table A.3 reports the 20 most frequent bigrams of our *Federalist Papers* corpus and Table A.4 the most frequent trigrams.

As for the POS tags, the number of single letters or characters is rather limited (let us say 26 + space = 27) and thus the number of bigrams ($27^2 = 729$) or trigrams ($27^3 = 19{,}783$) can be managed by the classifier. Of course, considering longer sequences implies managing larger amount of terms, for example, with 4-grams ($27^4 = 531{,}441$), 5-grams ($27^5 = 14{,}348{,}907$), or 6-grams ($27^6 = 3{,}874{,}420{,}489$). With such longer letter sequences, in addition to the memory management, the processing time is increased by a factor of 100 or more [211].

As letter $n$-grams do not constitute clearly interpretable linguistic items, one can represent texts by taking account of the syllables and their structure. From a phonologic point of view, such a strategy puts an emphasis on the patterns of sounds related to an author (or category), with the related prosody, meter, and rhythmic structure. The mapping between letter $n$-grams and syllables is however not without ambiguity and could be more or less difficult depending on the language [304]. For English (or French), such a correspondence between graphemes (spelling patterns) and syllables is difficult to establish automatically without errors.

As a simpler method trying to approximate the syllabic structure, one can generate stylistic features based on sequences of consonants and vowels. To indicate a consonant, the letter $C$ is used and $V$ for a vowel. This pattern could be repeated many times for a given word. A sequence of consecutive consonants is indicated by $C^n$ in which the value of $n$ is any integer value larger or equal to one and similarly for $V^n$. Optional parts of a pattern are enclosed in square brackets ([]). The sequence length is therefore not fixed, but it must respect one (or more) predefined pattern(s).

For example, the extraction of syllables could be based on the simple $C^n V^n [C^n]$ pattern, meaning a sequence of consonants followed by one or more vowels and optionally by a second sequence of consonants. For example, from the word "conference," the system will extract, as possible syllables, the sequences "con," "fe," "ren," "ce." This kind of stylistic representation is however not frequent in the domain.

Based on such word decomposition and with some adjustments,[5] the metric line or rhythm of a poem (or a play written in verses) can be identified. For example, Shakespeare tends to frequently write iambs, a metrical foot composed of two syllables with the first unstressed and the second stressed. When underlining the stressed parts, one can hear the "music" behind Shakespeare's expressions, for example, with "To be or not to be that is the question," an iambic pentameter (a sequence of five iambs).[6] Viewing Shakespeare as foremost a poet, the rhythm in his plays could be a fingerprint of his authorship (and even in plays written in prose) [302].

[5] Even if the word "compare" can be divided into three parts ("com," "pa," and "re") according to the previous model, it contains only two syllables "com" and "pare" because the sound of the final "-e" is ignored.

[6] As reasons justifying its frequent usage in plays, one can mention that this stylistic structure seems to be easier to remember for the actors and easy to follow by the audience with a music such as de-dum, de-dum, de-dum ... See the prosodic Python library https://www.gihub.com/quadrismegistus/prosodic.

Recently, different approaches have been suggested to represent more abstractly stylistic features. In this perspective, a text distortion model [136, 374] is taking account of only the $k$ most frequent word-types and by respecting their uppercase and lowercase letters. For the remaining tokens, each letter is replaced by an "*", each digit by a "#" and other symbols are kept as they are. In Table 5.1, two text distortion models are depicted, the first (labeled "Distortion $k = 0$") indicates only the length of both words and numbers and leaves the remaining symbols as they appear. When setting $k = 200$, the 200 most frequent word-types are added in the text surrogate as shown in the line "Distortion $k = 200$." After this text alteration, the learning stage considers variable length letter and word $n$-grams (e.g., $n = 1, 2$, and 3 for words and $n = 3, 4, 5, 6$ for letters) [374].

**Table 5.1** Examples of representations with text distortion or blenching techniques

| Text | Dinner | with | Paul, | 3 | beers | 👍, | I | was | 😃 | . |
|---|---|---|---|---|---|---|---|---|---|---|
| Distortion $k = 0$ | ****** | **** | **** , | # | ***** | 👍, | * | *** | 😃 | . |
| Distortion $k = 200$ | ****** | with | **** , | # | ***** | 👍, | I | was | 😃 | . |
| Frequency | 0 | 4 | 0 | 3 | 2 | 2 | 5 | 5 | 2 | 5 |
| Length | 06 | 04 | 05 | 01 | 05 | 02 | 01 | 03 | 01 | 01 |
| Shape | ull | ll | ullx | d | ll | jx | l | ll | j | x |
| CVC | cvccvc | cvcc | cvvco | o | cvvc | oo | v | cvc | o | o |
| Punctuation | w | w | w, | 3 | w | 👍, | w | w | 😃 | . |

In a similar vein, van der Goot et al. [412] propose a blenching approach to extricate a more abstract representation, less related to a given source language. Different variants have been suggested as shown in Table 5.1. First, the occurrence frequency of each word-type can be assigned to a given class. For example, words having a frequency between 0 and 5 are allocated to the class "0," while words depicting a frequency between 6 and 10 have the label "1," etc. In Table 5.1, this solution is displayed in the row "Frequency." Clearly, the words themselves do not appear in this surrogate, only their frequency class label. Assuming that the tag "5" corresponds to the most frequent word-types, the sequence "5 5" signals the occurrence of two very frequent words. According to the language, this could replace the bigram "of the" in English, "in der" in German, or "de la" in French.

As a second solution, one can represent each token by its length (the number of letters, digits, and symbols) as illustrated in the row "Length." To avoid confusion with the "Frequency" class labels, each value is preceded by a "0."

In a third strategy shown under the "Shape" line, an uppercase letter is transformed into a "u," lowercase into a "l," a digit by a "d," a punctuation symbol with an "x," and each emoji with a "j." Repetition of the same transformed character is limited to two to permit a better generalization. For example, the word *with* is replaced by "ll" and not "llll."

In a related mapping, under the row "CVC," each vowel is indicated with a "v," each consonant with a "c," and all other symbols by an "o." In the last row

("Punctuation"), each sequence of letters is replaced by a "w" and all other symbols are kept (because they are language-independent).

As for the text distortion approach, the final text representation is based on variable length letter and word $n$-grams (e.g., $n = 1$ and 2 for words, and $n$ between 3 and 6 for letters). Moreover, two or more text blenching strategies can jointly be applied to generate a more complex text surrogate.

Instead of focusing on words, and as discussed in Sect. 2.4, different vocabulary richness measurements can also be employed as stylistic features (e.g., the mean word length). The vocabulary as a whole is analyzed to generate a single value (e.g., the mean or the median), a set of values (e.g., min, max, mean, standard deviation), or a sample of values reflecting the underlying distribution. Such overall stylistic measurements tend to be not so effective for authorship attribution. However, one can take them into account for other applications (e.g., author profiling) as well as additional stylistic characteristics. Such overall measures have been found effective to discriminate between sets of tweets sent by a bot or a human [81, 222]. For example, machine-generated tweets tend to have a high lexical density (LD), or in other words, a large number of nouns, adjectives, verbs, and adverbs [177]. This pattern corresponds to, for example, job offers in which the determiners, propositions, or personal pronouns are simply missing (see the first tweet in Table 5.2 below). When applying the type–token ratio (TTR), one can observe that a series of tweets sent by computers are repetitive and thus present a high TTR value (few word-types for many tokens).

As another source of evidence about the veritable author or to identify a given category, one can look at the spelling errors, usually employed by forensics linguistics [291] or in a profiling application. For example, it is known that younger writers might present more spelling errors than older ones [213]. The geographical origin of the author could be detected by considering the spelling or spelling variants of some words. For example, Japanese or Chinese writers tend to forget to capitalize proper names and geographical entities (e.g., *obama*, *berlin*, or *spain*).

Depending on the text genre or support, other sources of evidence can be useful to identify the true author or the right target category. Some of them are more rarely used, such as the presence (or absence) of a greeting statement, a farewell sentence, paragraph indentation (as well as layout related to sections, chapters, footnotes, etc.). As the mean sentence length (MSL) can be useful to discriminate between several time periods, one can consider the number of sentences per paragraph as a possible stylistic marker. Of course, one must assume that all these possible variables are fully under the control of the author and are not imposed by an external source (e.g., the editor or publisher).

With the Web, various other stylistic characteristics could be pertinent to identify the author (or some of his demographics) or to verify whether or not the information is authentic. As an example, Fig. 5.1 displays a tweet[7] within its context.

[7]This tweet is a fake one (meme) generated from a dedicated website.

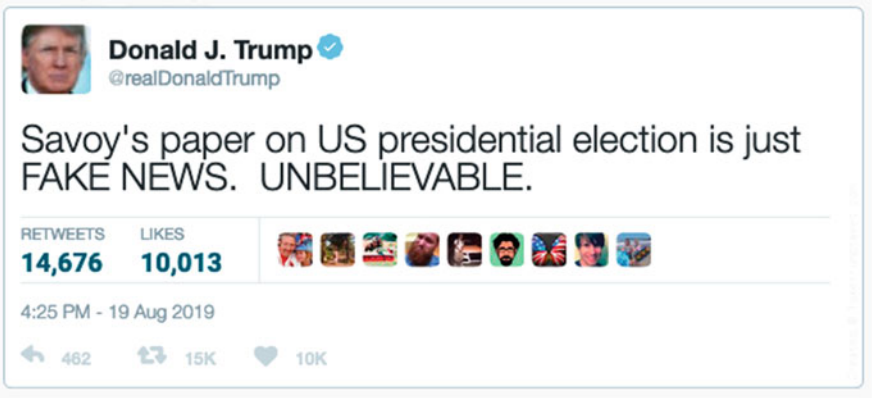


**Fig. 5.1** Example of a tweet with its surrounding context

As shown in Fig. 5.1, tweet users tend to write with uppercase letters to place emphasis on a given point or to indicate "I'm speaking aloud." Thus, the percentage of letters in uppercase or the number of words written entirely in uppercase letters could be considered as valuable stylistic markers.

Figure 5.1 also displays elements outside the text itself that can be useful for some specific text categorization applications. One can see the number of retweets (14,676) and the number of likes (10,013) associated with this tweet. These values could reflect the popularity of the sender or the public interest. Shu and Liu [359] propose to take account of them as additional characteristics to automatically detect fake news (see also [4]).

To specify the context of a short text message, hashtags (e.g., #nobama) appearing in a tweet indicate the general topic and could be useful to classify posts. Some of the most popular hashtags are #love, #fashion, or #photooftheday. Table 5.2 illustrates this with the hashtags #JOB, #hiring, or #health. As the Web is viewed as huge, hypertext, blog posts, tweets, Instagram posts, and other forms of e-message applications allow the sender to include mentions (e.g., @rogerfederer) or links (e.g., http://www.nasa.edu) to another post or a web page. The density of such items could be pertinent to discriminate between different categories. For example, younger persons tend to employ more links than elderly users.

**Table 5.2** Three examples of text appearing in tweets

| |
|---|
| #JOB #medical Mental Health Nurse https://www.t.co/i9PEEOOx2 #hiring #health https://www.t.co/HlAmnmpjPZ |
| RT @CuntsWatching: "No idea he cut hair" https://www.t.co/qAgs7rRGR3 |
| Diner tonight . . . Nettles a la crème https://www.t.co/eE4vycXV9h !!! |

In addition, chats, tweets, and other instant messaging offer new forms to shorten the message and to show the author's emotions. First, numerous acronyms and abbreviations in web-based communication correspond to new linguistic elements characterizing such media (e.g., "lol": laughing out loud, "omg": Oh my god, "pos": parents over shoulder) [77]. Allowing one to complete a message faster, this writing form is very popular, especially with younger people [256].

Second, instant messaging services have also adopted smileys, emoticons, or emojis[8] to allow the sender to indicate his emotions, sentiments, verbal tone, or the irony contained in a post [355]. These features partially denote the linguistic evolution [256]. It was recognized that the emotions present in messages are pertinent features to classify texts [313].

These particular features raise a representation problem. Does it make sense to include each specific hyperlink in a text surrogate. For example, with tweets depicted in Table 5.2, it is not clear whether or not the link "https://www.t.co/qAgs7rRGR3" must be included as it is in the text representation. One solution is to replace each hyperlink by a fixed form (e.g., *hyperlink*). Thus, only the presence or the number of such forms is taken into account. This issue appears also with hashtags or mentions. One can simply ignore the prefix symbol (e.g., # or @) and keep the following word. As a variant, all hashtags and addresses could be replaced by a fixed staring (e.g., *hashtag* and *mention*).

Extracting word-tokens from web-based posts is not always obvious. The notion of word seems to be more or less clear; a word is defined as a sequence of letters delimited by a space or a punctuation symbol. In a few cases, the answer is not fully clear. With "don't" or "Paul's," one can count one word (Paul or Paul's), two (Paul and s), or three (Paul, ', and s). A careful look at Table 5.2 reveals other problems. How many tokens do you see in the string "😭😭😭"? One or three? And in the sequence "!!!"? Different answers can be provided and it is not clear which one is the most appropriate.

Finally, some stylistic representations seem more adapted for a given medium or support such as a sequence of tweets. For example, in a recent PAN-CLEF evaluation campaign [314], the question to solve is to detect whether a set of tweets was written by a bot or a human and, in the latter case, whether these tweets have been written by a man or a woman. As a possible representation of a tweet, Kosmajac et al. [222] suggest to apply a DNA-based approach proposed by Cresci et al. [68]. In this view, the tweet surrogate is generated based on a set of binary questions. Table 5.3 depicts five of them. Given a single tweet, the system sums the values associated with a positive answer to this predefined set of questions. If a tweet contains one (or more) hashtag (value = 1), no mention, one URL (value = 4), no retweet, and 20 letters in uppercase (value = 16), the signature value for this tweet is $1 + 4 + 16 = 21$. Instead of retaining this number, the system adds 21 to

[8] Smileys are the oldest types (beginning in 1972), built in a round with two eyes and a mouth. As generalization, the emoticons (since 1982) are formed with punctuation marks, letters, and numbers (e.g., :-D). The emojis (1999) are pictographs of faces, objects, or symbols (e.g., 😭).

the code of the letter "A" (value 65 in the ASCII coding) and obtains the value 86 or the letter "V."

**Table 5.3** Examples of DNA-based representation

| Value | Condition |
|---|---|
| 1 | Has hashtags? |
| 2 | Has mentions? |
| 4 | Has URLs? |
| 8 | Has retweet? |
| 16 | Number of uppercase letters > 10 |

After applying this coding for the first tweet, one can repeat the same process for all tweets contained in a document and the resulting surrogate is a sequence of letters (e.g., VDE...). This surrogate generation can be generalized by changing the set of binary questions to more closely reflect another medium (e.g., percentage of personal pronouns larger than 4%). Second, instead of considering one tweet, one can choose to produce one letter for each sentence, each passage of two (or more) sentences, or even per paragraph. To represent the style of a given category, a variable length of letter $n$-grams is usually suggested (e.g., with $n = 3$ in [222]).

Finally, it is interesting to note that this representation technique has the advantage of taking account of the time or the sequence of the writing. Thus, one can compare texts starting or finishing with a similar sequence of letters.

## 5.3 Frequency-Based Feature Selection

As described previously, several sources could be pertinent to generate stylistic attributes. To achieve a high effectiveness level, the combination of different stylistic features is the norm rather than the exception. However, working with a large and sometimes a huge number of features renders both the learning and the assignment more complex. In this large attribute set, some of them are simply inaccurate or noisy because they are unable to provide useful indications about the correct target category. Such noisy features must be ignored and their removal usually improves the classification accuracy. In addition, some features are simply redundant and simply taking account of one of them is enough to discriminate between the different categories. Moreover, working with a reduced set of attributes will speed up the underlying computation and decrease the risk of overfitting the classifier to the available data [156].

Working with the *Federalist Papers*, Mosteller and Wallace [273] have defined a reduced list to 35 word-types to discriminate between articles authored by Hamilton or Madison (see Appendix A.3). This is clearly a large feature space reduction knowing that one can count 7994 distinct word-types appearing in this corpus.

As a first feature selection approach, and as described in Chap. 3, the number of features could be reduced by considering the 50–800 most frequent words (or lemmas) [46, 168]. When representing text with isolated words, the range between 200 and 500 tends to perform better [339]. As a variant, one can impose that useful features must display a relative occurrence frequency larger than a fixed threshold (e.g., larger than 0.05%). Of course, instead of considering isolated words, the same principle could be applied to word $n$-grams, POS sequences, or letter $n$-grams.

For some applications, the words could be selected according to their grammatical categories. For example, one can define the functional words as determiners, prepositions, pronouns (or limited to only personal pronouns), conjunctions, and modal and auxiliary verbs. When the target classifier must identify emotions (e.g., in a profiling task), the selected POS tags could be limited to adjectives (e.g., good, miserable, depressed) and manner adverbs (e.g., angrily, badly). When representing texts with POS sequences, constraints can be imposed to them, for example, collecting only sequences of adjective–noun–noun or noun–noun–noun trigrams as well as noun–noun bigrams.

In other cases, the selection could be based on predefined wordlists, for example, those present in the LIWC system [299, 389] designed to detect an author's psychological traits [40, 143]. Generating pertinent wordlists closely reflecting the underlying target class is always problematic and partially subjective (e.g., if *cry* or *tear* clearly denotes a negative emotion, what about *fake* or *protest*?).

As a variant, Jockers et al. [184] or Eder et al. [103] suggest applying a *culling* threshold to remote some selected terms. For example, when a single text represents 70% of the total frequency of a given feature, this term must be removed from the final feature set.

To choose the most appropriate subset of attributes, one can take account of the feature distribution among authors. As an extreme position, one can enforce that each useful term must appear, at least once, in a text written by each of the possible authors (or categories). The underlying idea is to consider only linguistic items known by all writers (or a given percentage of the authors). A term occurring with only one writer (e.g., *covfefe* tweeted by Trump on May 31, 2017) must be discarded [339]. Of course, such terms may be good indicators of the real author, but they are also easy to use by another person who wants to play a masquerade.

To demonstrate the effect of such feature selection approaches, one can return to the *Federalist Papers*. Within this corpus, one can count 7994 distinct word-types (vocabulary size with punctuation symbols). Enforcing that each word must appear under the pen of all possible authors the total number of word-types is reduced to 2907 words [183]. Finally, adding the constraint that each term must present a relative frequency higher than 0.05%, the number of features is reduced to 298 words. This example illustrates that the previous selection technique can be combined, for example, into a pipeline process.

## 5.4 Filter-Based Feature Selection

Instead of choosing a feature subset based on their frequencies, another paradigm suggests to ground the selection on the quality of each feature for *discriminating one category over the others*. The principle is to measure the fitness of each attribute over each individual category to define its *local utility* value. Because such fitness values are computed according to a single category, the adjective *local* is added. Then, based on values computed for each category, its overall discriminative power, or global utility value, can be derived. In a last step, all features are sorted, from the most pertinent to the less useful. The proposed subset is simply determined by extracting the top $m$ most discriminative features or those having a global score larger than a predefined threshold.

In text categorization [356], different local feature-scoring functions have been suggested. To measure the utility value of a term $t_i$ according to a given category (or author) $c_j$ (with $j = 1, 2, \ldots, r$) a contingency table can be generated as depicted in Table 5.4. In this table, the value $a$ indicates the number of texts belonging to the category $c_j$ in which the term $t_i$ occurs. When considering all other classes (denoted by $\bar{c}_j$), the term $t_i$ appears in $b$ other texts. Thus, in the whole corpus, this term occurs in $a + b$ documents, while we can count $a + c$ texts labeled with the category $c_j$.

**Table 5.4** Example of a contingency table for a term $t_i$ and a category $c_j$

| | Category $c_j$ | Category $\bar{c}_j$ | |
|---|---|---|---|
| Term $t_i$ | $a$ | $b$ | $a+b$ |
| Other terms ($\bar{t}_i$) | $c$ | $d$ | $c+d$ |
| | $a+c$ | $b+d$ | $n=a+b+c+d$ |

As a first local utility function between a term $t_i$ and a category $c_j$, one can compute the *pointwise mutual information* (PMI) given in Eq. 5.1 [57].

$$\begin{aligned} PMI(t_i, c_j) &= log_2\left(\frac{p(t_i, c_j)}{p(t_i) \cdot p(c_j)}\right) \\ &= log_2\left(\frac{\frac{a}{n}}{\frac{a+b}{n} \cdot \frac{a+c}{n}}\right) = log_2\left(\frac{a \cdot n}{(a+b) \cdot (a+c)}\right) \end{aligned} \tag{5.1}$$

This fitness function compares two models to estimate the probability of selecting the term $t_i$ for describing the category $c_j$. The first model is based on a direct estimation of the joint probability (denoted $p(t_i, c_j) = a/n$). This estimation is the numerator of Eq. 5.1. The second model (the denominator of Eq. 5.1) estimates this probability by considering independently the probability of the occurrence of the term $t_i$ ($p(t_i) = (a + b)/n$) and the probability of selecting a text belonging to the category $c_j$ ($p(c_j) = (a + c)/n$). When this assumption is true, the correct

estimation for $p(t_i, c_j)$ is $p(t_i) \times p(c_j)$. In this case, the two probability estimates will be close and the ratio described in Eq. 5.1 will return a value close to 1. Computing the logarithm of such a value, a value close to 0 indicates independence between the term occurrence and the corresponding category.

On the other hand, when a strong association does exist between the term $t_i$ and the category $c_j$, the value of $a$ will be large. The direct estimation for $p(t_i, c_j)$ will be larger than the product $p(t_i) \times p(c_j)$. The ratio will then be larger than 1 and the logarithm function returns a positive value. With an opposition between the term $t_i$ and the category $c_j$, the numerator will be smaller than the denominator, returning a value smaller than 1. Taking the logarithm, a negative value is provided, indicating that the term $t_i$ is less frequently used in category $c_j$ than in the rest of the corpus.

To illustrate this idea, a numerical example extracted from the *Federalist Papers* is depicted in Table 5.5. Clearly, the term *kind* appears in 35 articles written by Hamilton. The two other writers have employed the word *kind* in four other documents (to be precise, in three articles authored by Jay and one by Madison). The last row in Table 5.5 indicates that Hamilton wrote 51 texts, while the other categories regroup 19 texts (14 by Madison and 5 by Jay).

**Table 5.5** Contingency table for the word *kind* and the category "Hamilton"

| | Hamilton | Madison and Jay | |
|---|---|---|---|
| *kind* | 35 | 4 | 39 |
| not *kind* | 16 | 15 | 31 |
| | 51 | 19 | 70 |

A quick inspection of the values in this table confirms that the term *kind* is associated with Hamilton's articles. Hamilton wrote 51 papers in this corpus and one can find 35/51 = 68.6% of them with, at least, one occurrence of the word *kind*. For papers written by the two other writers, one can find only 4/19 = 21% of them with the term *kind*. Clearly, this word occurs more frequently in articles written by Hamilton.

The second example illustrates the distribution of the word *fully*. As displayed in Table 5.6, this term appears in 13 articles authored by Hamilton and in 12 others written either by Madison or Jay (more precisely, one can find 11 articles with the term *fully* written by Madison and one by Jay). This word occurs in only 25% (13/51) of Hamilton's papers compared to 63% (12/19) for the other writers. In other words, knowing that Hamilton's texts correspond to 73% (51/70) of the corpus, this word occurs less frequently 52% (13/25) in Hamilton's articles (instead of 73%).

According to values shown in Table 5.5, the PMI(*kind*, Hamilton) = $log_2((35 \times 70)/(39 \times 51)) = 0.3$. Such a score with a positive value signals an association between *kind* and Hamilton's style. With data presented in Table 5.6, the value of PMI(*fully*, Hamilton) = $log_2((13 \times 70)/(25 \times 51)) = -0.487$. A negative value implies that an opposition does exist between Hamilton's *lexis* preferences and the term *fully*.

**Table 5.6** Contingency table for the word *fully* and the category "Hamilton"

| | Hamilton | Madison and Jay | |
|---|---|---|---|
| *fully* | 13 | 12 | 25 |
| not *fully* | 38 | 7 | 45 |
| | 51 | 19 | 70 |

As one can deduce, the PMI function has the advantage to clearly indicate if an association between a feature and a category does exist or not. Moreover, the polarity of the association is also provided (positive with a value > 0 and an opposition with a value < 0).

As a second feature-scoring function, one can apply the *odds ratio* (OR) [249] that always returns a positive value. A positive association between the term $t_i$ and the category $c_j$ is indicated by a value larger than one, while a value close to zero signals an opposition. A value close to one denotes independence between the term and the underlying class.

$$\begin{aligned} OR(t_i, c_j) &= \frac{\frac{p(t_i|c_j)}{1-p(t_i|c_j)}}{\frac{p(t_i|\bar{c}_j)}{1-p(t_i|\bar{c}_j)}} = \frac{p(t_i|c_j) \cdot (1 - p(t_i|\bar{c}_j))}{(1 - p(t_i|c_j)) \cdot p(t_i|\bar{c}_j)} \\ &= \frac{\frac{a}{a+c} \cdot \left(1 - \frac{b}{b+d}\right)}{\left(1 - \frac{a}{a+c}\right) \cdot \frac{b}{b+d}} = \frac{a \cdot d}{b \cdot c} \end{aligned} \tag{5.2}$$

With data depicted in Table 5.5, the OR(*kind*, Hamilton) = $(35 \times 15)/(4 \times 16) = 8.2$, a large positive value signaling an association between the occurrence of the word *kind* and texts written by Hamilton. With the frequencies shown in Table 5.6, OR(*fully*, Hamilton) = $(13 \times 7)/(12 \times 38) = 0.2$, resulting in a small value, implying an opposition between Hamilton's style and the word *fully*. In other words, the presence of this term indicates that the underlying text is not authored by Hamilton. Does it help? Does it make sense to keep such terms? In fact, one prefers terms with a positive association, terms used frequently by one writer and ignored by the others. This comment does not imply that the term *fully* is not without merit. Knowing that this term appears in 11 articles written by Madison over a total of 14, OR(*fully*, Madison) = $(11 \times 42)/(3 \times 14) = 11$. A strong positive association is found between *fully* and Madison's style.

As a third local utility function, the *chi-square* $\chi^2(t_i, c_j)$ statistics [249] defined by Eq. 5.3 can be applied. The resulting value follows a chi-square distribution with four *dof* (degrees of freedom). Such values are always positive (whatever is the association between the feature and the category). A large positive value specifies a positive or negative association, while a small value signals independence. More precisely, with four degrees of freedom, the limit between having a significant relationship or not is 7.78. More precisely and according to this distribution, the chance of observing a value of 7.78 or larger is only 10%. Of course, one can be

more conservative and use the limit of 9.49 (5% of the distribution) or even 13.38 (1% of the possible values). When applying this selection function, one can have a theoretical limit to assess the significance of the relationship between the feature and the category.

$$\begin{aligned}\chi^2(t_i, c_j) &= \frac{n \cdot \left((p(t_i, c_j) \cdot p(\bar{t_i}, \bar{c}_j)) - (p(t_i, \bar{c}_j) \cdot p(\bar{t_i}, c_j))\right)^2}{p(t_i) \cdot p(\bar{t_i}) \cdot p(c_j) \cdot p(\bar{c}_j)} \\ &= \frac{n \cdot (a \cdot d - c \cdot b)^2}{(a + c) \cdot (b + d) \cdot (a + b) \cdot (c + d)}\end{aligned} \tag{5.3}$$

Using values depicted in Table 5.5, the $\chi^2$(*kind*, Hamilton) = 12.7 a large positive value. This observed value is clearly larger than the threshold of 9.49 (and values larger than this threshold correspond to only 5% of the distribution).

With the frequencies shown in Table 5.6, $\chi^2$(*fully*, Hamilton) = 8.55 denoting clearly a relationship between the category and the term *fully* (a value larger than 7.78). The chi-square value however does not directly provide the polarity of the association. In this case, one can observe (see Table 5.6) a negative association; the presence of the term *fully* in an article must be interpreted as this paper was not written by Hamilton.

Following the same interpretation as the chi-square function, one can employ the *information gain* (IG), also called the *expected mutual information*. The utility value returned by this function is always positive. A very small positive value implies the absence of a discriminative power for the term $t_i$ with respect to the category $c_j$.

$$\begin{aligned}IG(t_i, c_j) &= \sum_{c \in \{c_j, \bar{c}_j\}} \sum_{t \in \{t_i, \bar{t}_i\}} p(t, c) \cdot \log_2 \left( \frac{p(t, c)}{p(t) \cdot p(c)} \right) \\ &= \frac{a}{n} \cdot \log_2 \left( \frac{a \cdot n}{(a + b) \cdot (a + c)} \right) + \frac{b}{n} \cdot \log_2 \left( \frac{b \cdot n}{(a + b) \cdot (b + d)} \right) \\ &+ \frac{c}{n} \cdot \log_2 \left( \frac{c \cdot n}{(a + c) \cdot (c + d)} \right) + \frac{d}{n} \cdot \log_2 \left( \frac{d \cdot n}{(b + d) \cdot (c + d)} \right)\end{aligned} \tag{5.4}$$

Using values depicted in Table 5.5, the IG(*kind*, Hamilton) = 0.135 signifying a possible relationship between the feature *kind* and Hamilton's style. With data present in Table 5.6, IG(*fully*, Hamilton) = 0.086 tends to indicate a potential association between both the category "Hamilton" and the term *fully*. Unlike the chi-square function, the information gain does not provide any theoretical limit to assess whether or not a relation does exist. One needs to compare the values achieved with different features and then to rank them in decreasing order of their information gain value.

As a fifth function, the *gain ratio* (denoted GR) coefficient can sometimes be useful (see Eq. 5.5). As for the chi-square function, a positive or negative association

between the term $t_i$ and the category $c_j$ is indicated by a positive value, while a value very close to zero indicates independence.

$$\begin{aligned} GR(t_i, c_j) &= p(t_i, c_j) \cdot \log_2 \left( \frac{p(t_i, c_j)}{p(t_i) \cdot p(c_j)} \right) + p(\bar{t}_i, c_j) \cdot \log_2 \left( \frac{p(\bar{t}_i, c_j)}{p(\bar{t}_i) \cdot p(c_j)} \right) \\ &= \frac{a}{n} \cdot \log_2 \left( \frac{a \cdot n}{(a+b) \cdot (a+c)} \right) + \frac{c}{n} \cdot \log_2 \left( \frac{c \cdot n}{(c+d) \cdot (a+c)} \right) \end{aligned} \tag{5.5}$$

Using values depicted in Table 5.5, the GR(*kind*, Hamilton) = 0.037 a small positive value. As with the IG function, it is rather difficult to determine whether or not one can ascertain an absence of relationship (is this value very close to 0?). On the other hand, one can interpret this value as positive, denoting a positive or negation association. Moreover, the strength of the connection between the term *kind* and the texts written by Hamilton remains unknown. Based on data depicted in Table 5.6, GR(*fully*, Hamilton) = 0.025. Here too the interpretation is rather difficult.

As a sixth function, the GSS coefficient [126] signals a positive association with a positive value and an opposition with a negative value. When the returned value is close to 0, there is no relationship between the feature and the corresponding category. This coefficient is computed according to Eq. 5.6.

$$\begin{aligned} GSS(t_i, c_j) &= (p(t_i, c_j) \cdot p(\bar{t}_i, \bar{c}_j)) - (p(t_i, \bar{c}_j) \cdot p(\bar{t}_i, c_j)) \\ &= \frac{(a \cdot d) - (b \cdot c)}{n^2} \end{aligned} \tag{5.6}$$

Using values depicted in Table 5.5, the GSS(*kind*, Hamilton) = 0.094 a small positive value. It is rather difficult to determine whether one can conclude to an absence of relationship (because one can interpret such a value as close to 0). On the other hand, this value could be interpreted as positive. In this latter case, the association might be weak between the term *kind* and texts authored by Hamilton. As with other feature-scoring functions, a comparative basis is required to obtain a correct interpretation (see below). Based on data depicted in Table 5.6, GSS(*fully*, Hamilton) = $-0.07$ denoting an opposition between the category and the term *fully*.

Using one of these local selection functions, the possible features are ranked in decreasing order to their discriminative power for a given category. As an example, Table 5.7 depicts the top ten most useful terms for attributing an article either to Hamilton or Madison. For both possible authors, the values of the chi-square and the GSS coefficients are displayed together with the corresponding word-type. Appendix A.4 reveals a more complete picture with the top ten most discriminative terms for the six local utility functions considering both Hamilton's and Madison's styles.

In Table 5.7, all chi-square values are larger than the threshold of 13.38 (indicating that values larger than this limit represent only 1% of the distribution). For the GSS, no precise rule is provided specifying when a term should be selected

or not. Using a comparative basis, one can see that a large GSS value means larger than 0.07.

**Table 5.7** The top ten terms having the largest values for the category "Hamilton" or "Madison"

| Hamilton | | Madison | |
|---|---|---|---|
| Chi-square | GSS | Chi-square | GSS |
| 47.0 *upon* | 0.146 *upon* | 30.6 *whilst* | 0.094 few |
| 30.5 although | 0.094 kind | 28.9 upon | 0.089 particularly |
| 23.5 consequently | 0.083 intended | 25.7 composing | 0.089 *absolutely* |
| 23.1 wish | 0.082 community | 21.5 enlarge | 0.089 *whilst* |
| 19.9 whilst | 0.081 readily | 21.5 administering | 0.089 *proceedings* |
| 17.4 absolutely | 0.081 man | 21.5 indispensably | 0.089 consequently |
| 16.6 recommended | 0.076 matter | 21.5 assumed | 0.086 fully |
| 16.6 composing | 0.072 apt | 21.5 violating | 0.083 paper |
| 15.0 formed | 0.071 considerable | 21.5 *proceedings* | 0.077 trial |
| 14.9 paper | 0.070 enough | 21.5 *absolutely* | 0.077 during |

Of these six feature-scoring functions, which one is the most appropriate? The answer is not fully clear. From a theoretical point of view, it is rather difficult to favor one solution over the others. Each selection function owns a clear theoretical justification. In this choice, the target application might have an impact on the decision. Focusing on topical text categorization, Sebastiani [356] suggests to consider the chi-square or the GSS coefficient to rank the features according to their discriminative power. For Yang and Peterson [425] having the same objective, the most appropriate functions are either the odds ratio (OR) or the information gain (IG). Based also on empirical evidence, but focusing on the authorship attribution problem, Savoy [339] suggests ranking the features according to their term frequency (*tf*) or document frequency (*df*) or, as the second-best set of methods, one can apply either the gain ratio (GR), the chi-square, the GSS, or the information gain (IG) function.

As another way to compare those functions, one can analyze the degree of overlap between the terms extracted by those six functions. Based on the *Federalist Papers*, Table 5.7 shows that each function does not exactly favor the same set of words. For example, in the first two columns corresponding to Hamilton's style, the intersection between the GSS and chi-square approach is limited to a single word *upon*. When inspecting the two columns derived from Madison's style, the intersection counts three terms (*proceedings*, *absolutely*, and *whilst*).

When inspecting the percentage of terms in common when considering the top 50 most discriminative terms for both Hamilton and Madison (partially depicted in Appendix A.4), we observe that the PMI function is clearly distinct from all the others. The intersection with the five other functions is nil or very small (0.5%). On the other extreme, the chi-square function presents 76% of words in common with the IG solution or 59% with the GR approach. Of course, the terms in common between the function IG and GR are also relatively high (65%). The odds ratio (OR) proposes only a weak relationship with the GSS coefficient, having around 20% of words in common. The overlap with the other functions is less than 10%. In brief, four different groups of functions can be found, namely the trio composed of the chi-square, IG, and GR, in second the GSS coefficient, in third the OR, and finally the PMI function.

The single function without any empirical support is PMI. Even if this approach seems justified from a theoretical point of view, it tends to favor terms appearing only in one category, even when their occurrence frequency is low or even rare. For example, associated with Hamilton's class, PMI ranks in the first positions the terms *preface*, *aggression*, and *quadruple*. These three words never occur in articles written by Madison or Jay. In Hamilton's papers, one can find only one occurrence of *preface* and *quadruple* and two for *aggression*. To summarize, we do not suggest directly applying this term selection function because this approach favors rare terms. To partially solve this problem, Sect. 9.3 exposes how one can apply these functions in conjunction with other term selection approaches.

When applying one of the aforementioned selection functions, the focus is limited to one term and one category. When faced with a binary classification problem (two authors or two categories), such a local utility function is enough to define the overall selective value for each term. In stylometric studies, however, the number of authors (categories) is usually larger than two. In such cases one needs to aggregate the local utility values over the $r$ categories.

To define such a global utility measure for a term $t_i$, denoted $U_{op}(t_i)$, one can take the maximum over the $r$ categories (Eq. 5.7) or compute the sum (Eq. 5.8) or a weighted average as shown in Eq. 5.9. In the latter case, a weight $w_j$ is associated with each category to reflect its importance. Of course, the sum over all $w_j$ must be equal to 1. There is no theoretical or empirical justification clearly favoring one of these aggregation operators.

$$U_{max}(t_i) = \max_{j=1}^{r} f(t|c_j) \tag{5.7}$$

$$U_{sum}(t_i) = \sum_{j=1}^{r} f(t|c_j) \tag{5.8}$$

$$U_{wsum}(t_i) = \sum_{j=1}^{r} w_j \cdot f(t|c_j) \quad \text{with} \quad \sum_{j=1}^{r} w_j = 1 \tag{5.9}$$

An example is depicted in Table 5.8 based on the odds ratio function. For each of the three possible authors, the top ten most relevant terms are displayed. As each word is presented with a local utility value larger than one, this is a clear indication that the corresponding term could discriminate the corresponding class over the others. In the last column, the global utility is computed according to the *Max* operator. In this column, one can see two terms useful to identify Madison's style (*whilst* and *composing*) and the eight others are extracted from Jay's term set.

**Table 5.8** The top 10 terms having the largest values with the odds ratio (OR) and the fusion with the max operator

| Hamilton | Madison | Jay | Max |
|---|---|---|---|
| 17.9 there | 73.3 whilst | 96.0 firmly | 96.0 firmly |
| 16.0 intended | 55.0 composing | 48.0 nay | 73.3 whilst |
| 9.8 mentioned | 41.3 sphere | 47.3 productions | 55.0 composing |
| 9.8 commonly | 30.6 viewed | 47.3 gentlemen | 48.0 nay |
| 9.0 about | 30.6 pronounced | 47.3 invite | 47.3 productions |
| 8.2 kind | 30.6 relief | 47.3 continuing | 47.3 gentlemen |
| 8.2 matter | 30.6 respectively | 47.3 commodities | 47.3 invite |
| 7.6 apt | 30.6 planned | 42.7 compel | 47.3 continuing |
| 7.5 forward | 22.0 ten | 42.7 manage | 47.3 commodities |
| 6.3 community | 22.0 pieces | 42.7 provoke | 42.7 compel |

This example demonstrates a recurrent problem when selecting an appropriate subset of terms. Usually the final selection, whatever is the fusion operator, might favor one or two classes over the others (as shown in Table 5.8). The majority of chosen terms are worthwhile to discriminate Jay's style over the others. The three proposed aggregation functions do not include a factor favoring a fair selection *between* all $r$ categories.

One can design an ad hoc feature selection, for example, based on a round-robin strategy. In this case, one term is taken from each category during a round and this process is repeated until a given number of features has been reached. As a variant, a weighted round-robin can be implemented to take more terms for categories being more important or widespread.

In conclusion, these filter feature-scoring functions assess the importance of each feature independently of any learning schemes. Based on the feature distribution over the categories, these methods are therefore general. The computational cost is relatively low compared to wrapper methods presented in the next section. Such a fast execution is a real advantage when faced with a large number of features which is often the case with stylometric models. Filter methods tend however to select a relatively large subset of features and some of them could display a low frequency in the collection.

## 5.5 Wrapper Feature Selection

As a third paradigm to determine the best feature subset, the selection can take account of the impact of this subset on the overall effectiveness. The basic idea is then to evaluate all possible attribute subsets and to return the one achieving the highest performance. Such an iterative process implies that the learning scheme must be trained and tested for each possible subset to compute the resulting effectiveness.

This general strategy is possible when the number of features is rather limited. For example, to determine the optimal subset of six features over ten, the system must evaluate the classifier for 210 feature subsets (or $210 = 10!/(6! \times 4!)$). Considering six features is an arbitrary number of attributes. All other subset sizes must be evaluated, namely with a size $= 1, 2, 3, \ldots, 9$, up to 10. In total, 1,023 attribute subsets[9] must be generated, trained, and tested with the chosen learning scheme to discover the optimal one.

In stylometric studies, the number of stylistic markers is usually larger than ten. Even when limited to 200, the search space is clearly too large to be explored by an exhaustive search. For example, identifying the optimal subset of 50 attributes over 200 requires the generation and evaluation of 453,858,377,923,246,061,067,441,390,280,868,162,761,998,660,528 subsets (or $4.5 \times 10^{47}$).[10] Assuming one millisecond per evaluation and knowing that one century corresponds to $3 \times 10^{12}$ ms, this computation will require more than a billion centuries to complete the enumeration of all possible subsets. However, applying such a brute force strategy provides the main advantage to return the optimal feature subset, in conjunction with the chosen learning model.

Instead of adopting a blind and exhaustive search, a greedy approach can be applied. Adopting this point of view, two main strategies can be selected.

First, the step *backwards feature selection* seems to be the one most adapted with stylometric applications. In a first stage, the current solution is composed of all features. The classifier is built and the achieved performance defines the effectiveness of the current solution. The idea is then to explore all neighbors of the current state or all feature subsets that can be reached from the current one by a simple modification. To generate them, one feature at a time is removed from the current solution. If the starting subset contains $m$ attributes, $m$ neighbors can be generated. Iteratively, one by one, each feature is removed and the resulting performance is computed. Over all these $m$ neighbors, the best achieved solution is compared to the current one. If this reduction of one attribute does not improve the effectiveness, the current feature set is defined as the best one. Otherwise, the best neighbor is selected as the new current state. And the process restarts with this new current state by removing one by one a single feature. In this strategy, each

[9] With $m$ features, the number of possible subsets is $2^m - 1$ (minus one because we can ignore the empty set).

[10] Relatively close to the number of atoms in the earth estimated as between $10^{49}$ and $10^{50}$.

step decreases the number of features by one ultimately reaching an attribute subset having the highest performance.

As a variant to speed up the computation, one can change the current state as soon as one of the $m$ neighbors presents better performance. Thus, instead of exploring all $m$ neighbor subsets, only a fraction of them is generated and tested at each round.

Second, the *forward feature selection* works in the opposite direction, starting by considering only one feature. After evaluating the $m$ possible features, the best one forms the current state. Then, feature sets of size two are considered by adding a new attribute. If no solution is able to improve the current state, the current subset is identified as the best one. Otherwise, from all the generated subsets or neighbors, the one performing the best is taken as the current solution and the exploration continues by incrementing the feature size by one.

Applying the backward or forward greedy selection, there is no guarantee that the optimal feature subset is reached. This approach tends to produce an effective feature subset for the underlying classifier without any guarantee that this subset is also the best one for another learning scheme. Evidently, the computation cost is certainly the main disadvantage of this procedure. When faced with a relatively large number of features, the wrapper approach is computationally expensive even when applying a greedy approach (iterative exploration of neighbor solutions and selecting the best to continue the search).

## 5.6 Characteristic Vocabulary

Similar to the feature selection problem, the automatic identification of terms or expressions closely related to a category (or author) corresponds to a fundamental subject. After identifying such a reduced set of terms, one can explain, in plain English, an authorship attribution or justify an assignment to a given category. Just providing a very short answer such as "Hamilton" (the author's name or a category label) is too limited. For many people, it is rather difficult to trust such a black box system. Some arguments justifying the system's decision are needed to support the proposed assignment. Moreover, prior to applying a classification scheme, these methods can be useful to obtain an overall picture of the vocabulary characterizing each category (or author).

In our example, three possible authors are behind the *Federalist Papers* and could be the author[11] of one of the 12 disputed articles. They wrote with similar words or expressions but the differences between them reside mainly in their frequencies. To determine the terms overused (or underused) by an author, Muller [275] suggests analyzing the number of term occurrences between a writer compared to the whole corpus. For example, knowing that the word *upon* appears 370 times in Hamilton's speeches compared to 378 occurrences in the entire corpus, can we infer that

[11] As shown in Table 2.1, the uncertainty is only between Hamilton and Madison.

*upon* is overused (denoted C+), underused (C−), or employed with a non-different frequency (C=) than the other two writers?

To provide an answer to this question, the size (number of tokens with punctuation symbols) of the entire corpus is indicated by the variable $n$ (= 175,781), while the sample of texts written by Hamilton is denoted $n_1$ (= 123,125). For the chosen term (e.g., *upon*), one can count its number of occurrences in the entire corpus (value denoted by *tf* or 378 in our example). In a similar way, its absolute frequency in Hamilton's texts (indicated by $tf_1$ or 370) is calculated. According to this model, one can estimate the occurrence probability of the word *upon* in the whole corpus as $tf/n$ (or $378/175{,}781 = 0.00215$). Knowing that Hamilton's articles contain 123,125 tokens, one can expect observing, in average, $n_1 \times (tf/n)$ times the term *upon* in Hamilton's texts (or $123{,}125 \times (378/175{,}781) = 264.8$). This mean is clearly lower than the observed occurrence number (370).

More formally, assuming that the frequency for this term (denoted $\omega$) in Hamilton's sample is the same as for the other writers (our null hypothesis denoted $H_0$), one can estimate the probability that $\omega$ appears $tf_1$ times in Hamilton's speeches according to a hypergeometric law [14, 15] described by the following equation:

$$p(\omega = tf_1) = \frac{\binom{tf}{tf_1} \cdot \binom{n-tf}{n_1-tf_1}}{\binom{n}{n_1}} \tag{5.10}$$

In this formulation, the denominator indicates the number of ways one can select $n_1$ items from a larger set containing $n$ items, discarding order (binomial coefficient). The numerator takes account of two parts. First, the number of ways to choose $tf_1$ elements from a set comprising $tf$ items (the word *upon* in our example). Second, one needs to select the remaining $n_1 - tf_1$ items from the set having $n - tf$ elements (all other tokens in our example).

Estimating the probability of a single occurrence number only by Eq. 5.10 is not fully pertinent (and would be very small). One can admit that some natural variability does exist; one writer might employ this word a little bit more or less, without depicting a statistically significant deviation. To define the thresholds determining the limits of this variability, one can compute the cumulative distribution of the occurrence frequency of the term $\omega$ in the underlying corpus as indicated in Eq. 5.11.

$$p(\omega \leq tf_1) = \sum_{f=0}^{f=tf_1} p(\omega = f) \tag{5.11}$$

where $f$ represents the absolute frequency ($f = 0, 1, 2, \ldots, tf_1$) of the chosen word $\omega$ in Hamilton's sample. The maximal frequency is $tf$, the number of occurrences in the entire corpus. To apply the statistical test, one can define the lower and upper frequency limits (confidence interval) for which the cumulative probability

distribution reaches a specified threshold denoted $\alpha$ and $1-\alpha$ (e.g., $\alpha = 5\%, 1\%$, or even $0.5\%$).

For example, according to the null hypothesis $H_0$ and specifying $\alpha = 5\%$, one can observe between 250 and 279 occurrences of *upon* in Hamilton's speeches (assuming that this word-type is used with the same intensity under all authors). However, *upon* occurs 370 times in Hamilton's articles, too many occurrences to be inside the confidence interval. Thus, Hamilton employed the term *upon* more frequently and this word-type appears in his C+ vocabulary.

Using this technique, one can determine some words characterizing Hamilton's writings such as *upon, there, to, would, courts, of, court, jury, kind, in, an*, and *community*, all belonging in C+. For Madison, one can find *powers, confederation, department, on, Congress, articles, coin, legislative, executive, by, democracy*, and *constitution*. In the five articles written by Jay, the following list of characteristic terms can be found: *treaties, and, confederacies, nations, obeyed, gentlemen, fleet, America, wise, secrecy, tides*, and *they*. With each list associated with each author, one can observe a mixture of functional terms and content-bearing ones or a blend of stylistic and topical words [296, 337].

Instead of using Eq. 5.10 (a model based on a single urn), one could take account of the proportion of different POS categories, namely: nouns, verbs, adjectives, adverbs, and functional terms (e.g., prepositions, conjunctions, determiners, and pronouns). Thus, the different occurrences are no longer drawn from the entire corpus (or $n$) but from the number of tokens belonging to the corresponding POS category in the entire corpus ($n_{pos}$) or in the target subset ($n_{pos1}$). This proposed model now considers several distinct urns, one per chosen POS category.

$$p(\omega = tf_1) = \frac{\binom{tf}{tf_1} \cdot \binom{n_{pos}-tf}{n_{pos1}-tf_1}}{\binom{n_{pos}}{n_{pos1}}} \tag{5.12}$$

When considering the nouns, verbs, and functional categories and applying Eq. 5.12, the overused terms for the three authors of the *Federalist Papers* are depicted in Table 5.9.

**Table 5.9** The top five terms overused by three writers according to the grammatical categories nouns, verbs, and functional terms

| Hamilton | | | Madison | | | Jay | | |
|---|---|---|---|---|---|---|---|---|
| Noun | Verb | Funct. | Noun | Verb | Funct. | Noun | Verb | Funct. |
| court | proposed | upon | department | propose | on | treaty | composed | and |
| jury | constitute | an | confederation | composing | the | nation | obeyed | they |
| matter | intend | of | form | exercise | by | affair | joined | we |
| kind | be | he | Congress | regarded | among | America | recommend | although |
| thing | compose | in | constitution | report | these | fleet | deceive | or |

As a second method to define the vocabulary characteristic of a given category, one can apply the *tf idf* approach. In information retrieval and in automatic summarization, the selection of the best terms for describing a document (or a text sample) is usually based on the *tf idf* values [151, 249]. The idea is to rank the terms according to their intrinsic discriminative value or informative quality. To achieve this, one can consider that words depicting a high occurrence frequency can closely reflect the text compared to the other ones appearing less often in the same corpus. Therefore, words showing a high term frequency (or $tf$) are the most relevant. This simple strategy is not very effective as a very similar set of terms will be proposed for each document or sample of texts, those that are very frequent as shown in the Zipf's law (see Sect. 2.3).

Therefore, a second component must be taken into account. A useful term must be both frequent in a sample of texts and relatively rare in the rest of the corpus. This second criterion is reflected by the *idf* value [369] computed as the natural logarithm of the ratio between the number of documents in the corpus (indicated by $n$) and the number of documents where this term appears (denoted *df* for document frequency). When a term appears in a single text, its *df* value is one and the corresponding *idf* reaches is maximal value (e.g., $idf = ln(n/1) = ln(n)$). As the *Federalist Papers* count 70 articles, the maximal value is 4.25 (or $ln(70)$). On the other hand, when a word occurs in all texts, its *idf* value is minimal and reaches the value 0 (e.g., $idf = ln(n/n) = ln(1) = 0$).

To reflect these two components, the weight of the $i$th term in the $j$th document (or sample of texts) is denoted by $w_{i,j}$ and computed according to Eq. 5.13. This formulation takes account of the frequency of the term ($tf$ part) and its distribution over all documents (or all text samples) in the corpus (*idf* component).

$$w_{i,j} = tf_{i,j} \cdot idf_j = tf_{i,j} \cdot \ln\left(\frac{n}{df_j}\right) \tag{5.13}$$

As a first example of the *tf idf* extraction effect, Table 5.10 reports the top six word-types having, over all the corpus, the largest values. The selected terms reflect some of the topics of this corpus. When inspecting the $tf$ values, one can observe that those terms have a relatively high frequency. Moreover, these word-types appear in many articles written by the three possible authors.

When considering each author as a sub-corpus on its own, one can discriminate some of the subjects between the three authors as depicted by Table 5.11.

As the $tf$ factor is an integer (0, 1, 2, ...) and *idf* a real value between 0 and $ln(n)$, some applications suggest to separately normalize both components to achieve a value between 0 and 1. In this case, one can divide each $tf$ value by the maximal $tf$ value for a given document. With the *idf* part, the solution is simpler and one can divide it by the largest possible value, namely $ln(n)$. Equation 5.14 presents a normalized *tf idf* value.

**Table 5.10** The top six terms with the largest *tf idf* values and their distributions over the three authors

| | Overall | | | Hamilton | | Madison | | Jay | |
|---|---|---|---|---|---|---|---|---|---|
| Word | *tf idf* | *df* | *tf* | *df* | *tf* | *df* | *tf* | *df* | *tf* |
| executive | 168.45 | 33 | 224 | 22 | 127 | 9 | 94 | 2 | 3 |
| courts | 168.41 | 18 | 124 | 12 | 115 | 4 | 5 | 2 | 4 |
| jury | 158.34 | 8 | 73 | 5 | 70 | 3 | 3 | 0 | 0 |
| president | 143.27 | 21 | 119 | 15 | 97 | 5 | 12 | 1 | 10 |
| " | 143.15 | 45 | 324 | 33 | 211 | 9 | 101 | 3 | 12 |
| senate | 143.12 | 25 | 139 | 18 | 112 | 6 | 18 | 1 | 9 |

**Table 5.11** The top six terms with the largest *tf idf* values per author

| Hamilton | | Madison | | Jay | |
|---|---|---|---|---|---|
| *tf idf* | Word | *tf idf* | Word | *tf idf* | Word |
| 162.57 | jury | 44.63 | " | 16.09 | president |
| 126.63 | trial | 41.53 | executive | 14.48 | senate |
| 118.71 | president | 38.82 | department | 11.27 | her |
| 116.64 | senate | 33.08 | faction | 11.27 | congress |
| 106.78 | executive | 30.89 | majority | 11.27 | making |
| 166.40 | courts | 30.78 | confederation | 11.00 | made |

$$w_{i,j} = ntf_{i,j} \cdot nidf_j = \frac{tf_{i,j}}{\max_k tf_{k,j}} \cdot \frac{\ln\left(\frac{n}{df_j}\right)}{\ln(n)} \tag{5.14}$$

When comparing the rank lists of terms derived with the *ntf nidf* formulation, the difference with the classical *tf idf* version is rather small. For example, when analyzing the overall *Federalist* corpus, one can find a single difference with the six top ranked words reported in Table 5.10. The word *jury* is replaced by *her* when using the *ntf nidf* method. Similar conclusions are reached with each of the three authors. For example, the pronouns *his* and *her* appear higher in the list related to Hamilton's writings with the *ntf nidf* approach.

Of course, other implementation variants are possible, for example, by taking the logarithm of basis 10. Moreover, instead of considering isolated words, one can compute the values for a term in general (word $n$-grams, letter $n$-grams, POS sequences, etc.). The main drawback with the *tf idf* approach is the absence of any theoretical argument indicating when a word or an expression is significant or not for a given category [343].

# Chapter 6
# Machine Learning Models


Each quantitative linguistics or stylometric study can follow the following roadmap subdivided into six main steps. First, a precise research question or hypothesis must be formulated according to a theory or to verify a hypothesis. Second, a sample of texts must be collected to create an evaluation corpus [29]. Third, some preprocessing procedures must be applied to control the data quality, to remove extra-textual items (running titles, page numbers, etc.), and usually to normalize the spelling (one meaning = one spelling). Fourth, as discussed in Chap. 5, one text representation strategy must be chosen to extract pertinent stylistic features reflecting a category or a personal writing style. Of course, this selection depends on the target application. Fifth, as described previously, several feature selection procedures could then be employed to remove noisy attributes, to reduce the computational cost, and to decrease the risk of overfitting. In the sixth stage, a machine learning model is chosen and applied to the dataset. After learning a representation for each possible category (or author), the classifier should compute a distance (or similarity) between each class or instance representation and the disputed text surrogate. The most probable category (or author) is selected according to the smallest distance (or largest similarity) with the disputed text representation. In this computation, and depending on the classifier, each text could be individually represented (instance-based) or all texts belonging to a given category (or written by the same author) can be concatenated to create a single profile per category.

Before presenting the different machine learning models, we need to specify the terminology used in a domain shared by statisticians, humanity scholars, computer or data scientists. For a statistician, a model links several independent or explanatory variables to a dependent one, the label of the classification problem. Persons working on machine learning prefer talking about features to be combined by a model to predict a response. Computer scientists prefer to speak about different attributes employed to determine a given category or class. Of course, when solving a stylometric or authorship attribution question, terms (isolated words, *n*-gram



J. Savoy, *Machine Learning Methods for Stylometry*,
https://doi.org/10.1007/978-3-030-53360-1_6

of words or letters, etc.) are viewed as features or attributes to identify stylistic idiosyncrasies of a given author [80].

As describing all possible machine learning models is impossible, this chapter presents four basic approaches. The rest of this chapter is structured as follows: Section 6.1 describes the *k*-nearest neighbors (*k*-NN) and its variants. Section 6.2 explains the naïve Bayes model applied to stylometric problems. In Sect. 6.3 the support vector machine (SVM) approach is discussed, one of the most effective approaches for solving text categorization problems. Section 6.4 presents a second successful learning scheme, the logistic regression model. The last section exposes how to apply and evaluate these models with R using the *Federalist Papers* corpus as an example.

## 6.1 *k*-Nearest Neighbors (*k*-NN)

The nearest neighbor (NN) model represents each instance as a vector (or a point) in a *m* dimensional space where each dimension corresponds to a feature (or a term). As an example, Table 6.1 shows the surrogate of eight *Federalist* articles into four dimensions, namely the prepositions *by*, *upon*, *on*, and the modal verb *would*. The three possible authors appear with two texts and the last two columns depict the representation of two disputed articles, namely Q54 and Q55. In this table, each cell indicates the percentage of tokens occurring in the corresponding article. In article H8, for example, one can see 0.55% of the tokens corresponding to the preposition *by* and 1.36% to the modal verb *would*. As these percentages indicate the relative frequencies, one can interpret them as an estimation of their occurrence probability.

**Table 6.1** Percentage of four selected word-types in newspaper articles written by three authors and in two disputed articles

| Term | H8 | H59 | M38 | M40 | J2 | J64 | Q54 | Q55 |
|---|---|---|---|---|---|---|---|---|
| *by* | 0.55% | 0.90% | 1.11% | 1.82% | 0.60% | 1.30% | 1.30% | 0.69% |
| *upon* | 0.15% | 0.16% | 0.12% | 0.00% | 0.06% | 0.00% | 0.10% | 0.00% |
| *on* | 0.45% | 0.32% | 0.45% | 0.43% | 0.48% | 0.61% | 0.95% | 0.44% |
| *would* | 1.36% | 0.84% | 0.45% | 0.20% | 0.30% | 0.30% | 0.30% | 0.49% |

An important preprocessing step with the NN model is to ensure that each attribute is measured in the same or very similar units. Looking at the data depicted in Table 6.1, one can assume that this is really the case. In other contexts, the dimensions used for the different attributes could be very dissimilar. Let us take a second example with as attributes the absolute frequency of the determiner *the* and the preposition *upon* measured in number of tokens as shown in Table 6.2.

In this table, Madison's paper M38 has a length of 3319 tokens in which one can count 269 times the determiner *the* and 4 times the preposition *upon*. The disputed paper Q54, with a length of 1996, contains 202 occurrences of *the* and two of *upon*.

When computing the distance between these two text representations based on the word-types, the frequency of the determiner *the* dominates the frequency associated with the preposition. Even when the two attributes are measured on the same scale (e.g., number of tokens), the large difference observed with the first attribute hides differences that can be observed with the second. For example, the Manhattan distance between M38 and Q54 is $|269 - 202| + |4 - 2| = 67 + 2 = 69$. Clearly, the frequency difference of the preposition plays only a marginal role compared to the difference with the determiner.

**Table 6.2** Length (number of tokens) and frequency of *the* and *upon* in papers written by three authors and in two disputed articles

| | H8 | H59 | M38 | M40 | J2 | J64 | Q54 | Q55 |
|---|---|---|---|---|---|---|---|---|
| Size | 1988 | 1899 | 3319 | 3014 | 1665 | 2306 | 1996 | 2043 |
| *the* | 155 | 175 | 269 | 292 | 105 | 172 | 202 | 180 |
| *upon* | 3 | 3 | 4 | 0 | 1 | 0 | 2 | 0 |

To assign to all attributes a similar impact in the intertextual distance, one needs to normalize them. For example, in Table 6.1, the relative occurrence frequency has been used instead of the absolute frequency. In general, various other functions could be employed to obtain this effect. Having $n$ instances in a sample, the current attribute value denoted $a_i$ can be replaced by its normalized value denoted $a'_i$ according to Eq. 6.1. In this variant, for a given predictor, each value is divided by the sum over all $n$ values.

$$a'_i = \frac{a_i}{\sum_{j=1}^{n} a_j} \tag{6.1}$$

Another normalization approach considers the minimal and maximal value for a given attribute (denoted by $min(a_j)$ and $max(a_j)$ in Eq. 6.2). Based on them, one can subtract the minimum for each attribute value and then divide the result by the range of possible values (or $max(a_j) - min(a_j)$). With this procedure, the minimal value will be replaced by a 0 and the maximal one by 1. All other values will appear between these two limits.

$$a'_i = \frac{a_i - min(a_j)}{max(a_j) - min(a_j)} \tag{6.2}$$

As another example, the Delta model standardizes all values to obtain their corresponding Z score as described in Sect. 3.1 or by Eq. 6.3. In this formulation, for each value, we subtract the arithmetic mean ($mean(a_j)$) and divide the result by the standard deviation ($sd(a_j)$). The standardized values have a zero mean and a unit

variance.[1] When the underlying data follows a Gaussian distribution, one can expect observing values between $-3$ and $+3$ and the majority of them will occur between $-1$ and $+1$. When choosing this method, one must ensure that the selected distance measure works with negative values (which is not always the case). To avoid this difficulty, one can add to all standardized values the maximal value ($max(a'_j)$) to generate only positive results.

$$a'_i = \frac{a_i - mean(a_j)}{sd(a_j)} \tag{6.3}$$

After normalizing the dataset, the learning model can be applied. The NN approach does not generate a specific learned representation but directly utilizes the instances belonging to the training set to compute the proposed assignment. One can view this strategy as a lazy learning approach. The essential aspect in the NN model is to define an appropriate distance function. However, no clear and conclusive theory can help us by suggesting one unique best distance (or similarity) function.

When limiting the number of attributes to two, one can clearly visualize the articles and the underlying distances between them. In Fig. 6.1, the horizontal axis indicates the relative frequency of the word *by* and *would* is represented by the vertical one. In this scatterplot, the six texts written by three authors are inserted according to values depicted in Table 6.1. Within this small training set, the two disputed articles (Q54 and Q55) are then added. Taking as example Q55, one can compute its distance to all texts belonging to the training set to determine its closest neighbor. To achieve this, the Euclidian function seems the unique choice.

But as discussed in Sect. 3.1, the Delta model suggests another distance function, or one can prefer Labbé's intertextual distance (Sect. 3.3), or even the Kullback–Leibler divergence described in Sect. 3.2. As numerous distance functions are available (45 are described in [52]), one of them can be selected according to some justifications.

To achieve this goal, some theoretical properties can be useful in characterizing a distance function and one of them is the *metric* property. To own this quality, a distance function between two vectors must respect the following four constraints:

1. Zero distance: when two vectors are identical, their distance must be zero.
2. Positive distance: when two vectors differ, the distance must be positive.
3. Symmetry: computing the distance from vector A to B must return the same value as computing the distance from B to A.
4. Triangle inequality: the distance between A and B must be smaller or equal to the distance between A and C plus the distance C and B, for any point C.

All four constraints are reasonable and many distance functions respect them. One can call them a *distance metric* or, in short, *distance*. As a counter-example,

[1] This procedure appears as argument is some R functions under the name `scale=TRUE`.

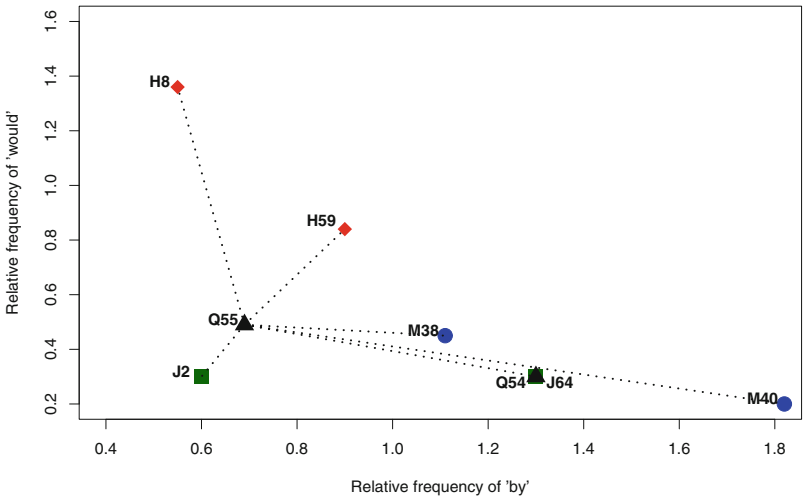


**Fig. 6.1** k-NN model with two attributes *by* and *would*

one can verify that the KLD function (see Eq. 3.3) is not a metric because it does not respect the symmetry property. But the name indicates this; KLD means Kullback–Leibler divergence and not distance.

As a first set of distance function, one can consider member of the $L^1$ family. Following the Delta and Labbé's method and as shown in Sect. 3.4, one can opt for the Manhattan formulation given in Eq. 3.8 in which only the amplitude of the absolute differences is taken into account.

Different variants of the Manhattan distance have been proposed by applying different normalization schemes. As an example, Eq. 6.4 specifies the Tanimoto distance in which the variable $a_i$ indicates the value of the $i$th component (or term) inside vector A (and similarity $b_i$ for vector B). In this formula, $m$ indicates the number of components of vector A or B or, in other words, the number of stylistic features.

With this formulation, the Manhattan distance (the numerator of Eq. 6.4) is normalized by the sum of the maximum value between $a_i$ and $b_i$. When adopting this computation, it is usually assumed that each component $a_i$ and $b_i$ have only positive values.

$$D_{Tanimoto}(A, B) = \frac{\sum_{i=1}^{m} |a_i - b_i|}{\sum_{i=1}^{m} max(a_i, b_i)} = \frac{\sum_{i=1}^{m} (max(a_i, b_i) - min(a_i, b_i))}{\sum_{i=1}^{m} max(a_i, b_i)} \tag{6.4}$$

The Canberra distance function is another example. Each component is normalized by the sum of two coefficients as shown in Eq. 6.5. Of course, as the sum is

over the $m$ attributes, the final value could be larger than one for both the Tanimoto and Canberra function. Other variants can be found in [209].

$$D_{Canberra}(A, B) = \sum_{i=1}^{m} \frac{|a_i - b_i|}{|a_i| + |b_i|} \tag{6.5}$$

All these $L^1$ functions are distances because they respect the four theoretical properties. It is still difficult to favor one of them. As they usually return different values, the choice is not evident. One can however look at the computational details. Consider the pair H59 and Q55 in Table 6.1. For the word *by*, the Manhattan distance is $|0.9 - 0.69| = 0.21$, while for *upon*, it is $|0.16 - 0.0| = 0.16$. Both values seem to be in the same range. Replacing the Manhattan function with the Canberra, we obtain for *by* $|0.9 - 0.69|/(|0.9| + |0.69|) = 0.21/1.59 = 0.132$ and for *upon* $|0.16 - 0|/(|0.16| + |0|) = 0.16/0.16 = 1$. Clearly, the Canberra function could largely penalize the absence of a feature in a text representation. This aspect could be important for some applications, but not for all. One can also mention that having a large number of features increases the probability that one of them will appear with a zero value in some text representations. Its absence will have a larger impact on the Canberra formulation.

As a variant of the Tanimoto formulation, one can apply the MinMax similarity measure defined by Eq. 6.6 and found effective by Koppel and Seidman [214]. When vector A is the same as B, all min or max operations return the same value. Therefore, the similarity value reaches 1. The minimal value of 0 is obtained when for all pairs of values $(a_i, b_i)$ we always have one of them equal to zero. To define a distance measure, one can simply subtract this similarity value from 1 as shown in Eq. 6.7.

$$Sim_{MinMax}(A, B) = \frac{\sum_{i=1}^{m} min(a_i, b_i)}{\sum_{i=1}^{m} max(a_i, b_i)} \tag{6.6}$$

$$D_{MinMax}(A, B) = 1 - Sim_{MinMax}(A, B) \tag{6.7}$$

Distance functions based on the $L^2$ norm have another starting point; large differences must account for more than small ones when computing an overall distance value as shown previously with the Euclidian distance (see Eq. 3.9).

When using this Euclidian distance with vectors depicted in Fig. 6.1, the closest distance (0.22) is between Q55 and J2, and the second closest is with M38 (0.44).[2] Applying the Manhattan function, the same ranking is achieved. However, selecting the Canberra or Tanimoto function instead of the Euclidian one, the closest point will be J64 (1.13 with Tanimoto[3] vs. 1.59 for J2). For all these functions, the

[2] In the dedicated webpage, an Excel sheet is available with this computation.

[3] Under the assumption that 0/0 = 0.

returned value is positive but does not have a precise maximum value. A meaningful interpretation of a value larger than 0 is therefore problematic. We are never sure if such a value is really small, rather small, in the mean, large, or huge. A comparative basis is mandatory.

As for the $L^1$ family, the $L^2$ norm offers many different functions, for example, the Matusita depicted in Eq. 6.8 achieving good empirical performance as demonstrated in [209].

$$D_{Matusita}(A, B) = \sqrt{\sum_{i=1}^{m} \left(\sqrt{a_i} - \sqrt{b_i}\right)^2} \tag{6.8}$$

As a third paradigm, one can consider a distance measure based on the inner product, also called dot product given in Eq. 3.10. The cosine function described in Sect. 3.4 (see Eq. 3.11 (similarity) and Eq. 3.12 (distance)) is a well-known example of this family.

As a second distance function based on the inner product, one can mention the Jaccard similarity shown in Eq. 6.9. This similarity coefficient could be applied with sets. In this case, the Jaccard similarity is the ratio between the number of elements in the intersection of the two sets A and B, divided by the size of their union. This similarity returns a value between 0 (nothing in common) and 1 (the two sets are identical). Therefore, the interpretation of the returned value is clear. When faced with weighted features, the similarity between two vectors A and B is defined by Eq. 6.10. As for the set version, this function always returns a value between 0 and 1.

$$Sim_{Jaccard}(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{6.9}$$

$$Sim_{Jaccard}(A, B) = \frac{\sum_{i=1}^{m} a_i \cdot b_i}{\sum_{i=1}^{m} a_i^2 + \sum_{i=1}^{m} b_i^2 - \sum_{i=1}^{m} a_i \cdot b_i} \tag{6.10}$$

Computing a similarity coefficient is different from a distance. To achieve this second result, one can simply apply Eq. 6.11, subtracting the similarity coefficient from one.

$$D_{Jaccard}(A, B) = 1 - Sim_{Jaccard}(A, B) \tag{6.11}$$

To complete the inner product-based functions, one can also apply the Dice similarity coefficient defined for sets by Eq. 6.12 and in Formula 6.13 for numerical vectors. As for the Jaccard distance, the Dice distance value is computed by subtracting the similarity coefficient from one.

$$Sim_{Dice}(A, B) = \frac{2 \cdot |A \cap B|}{|A| + |B|} \tag{6.12}$$

$$Sim_{Dice}(A, B) = \frac{2 \cdot \sum_{i=1}^{m} a_i \cdot b_i}{\sum_{i=1}^{m} a_i^2 + \sum_{i=1}^{m} b_i^2} \tag{6.13}$$

As a fourth paradigm, an entropy-based distance function can be proposed, for example, the KLD formula (see Eq. 3.3). Clearly, this is not a metric because the KLD expression does not respect the symmetry constraint.

In conclusion, several distance functions can be applied and, for each of them, different variants can be implemented. With the Delta model, for example, Evert et al. [111] describe and explain several variations (e.g., applying a Euclidian or a cosine distance function) within the framework proposed by Burrows [46]. As another example, Hoover [169] also suggests using the cosine distance with the Delta model (without achieving significantly better effectiveness).

Moreover, when selecting a distance measure to compute the distance between two stylistic representations, theoretical justifications do not provide a clear guide. Based on the $L^1$ norm, the Manhattan function is simple to compute. From a performance point of view, Kocher and Savoy [209] demonstrate that some variants of the $L^1$ family (e.g., Tanimoto) tend to provide higher effectiveness in several author profiling tasks. However, a complete answer to the question "Which is the best distance function for a given stylistic task?" remains elusive. This issue is related to the *no free lunch* theorem [422, 423]. When averaging the effectiveness over all possible problems, every classification algorithm has a similar accuracy rate when classifying new unseen data. In other words, no learning scheme can be universally better than all the others.

After choosing a distance function, the classifier could be based on the 1-NN method. As illustrated previously, the assigned attribution for a query text is the same as *the* closest neighbor in the training sample. To determine this one, the distance with all instances in the training sample must be computed, and the smallest indicates the closest neighbor.

Limiting the decision to a single instance could be problematic, especially when the dataset could be noisy.[4] Moreover, an effective classifier must be able to generalize the information provided by the training set. Taking a decision on a single observation could imply a risk of overfitting the results on the training data. Therefore, it is recommended to consider not a unique neighbor but $k$ of them.

When applying a $k$-NN approach, the distance of $k$ closest instances in the training sample is taken into account for providing the final assignment. When faced with two categories, $k$ is usually an odd number and the majority class found in the $k$-NN set determines the result. In the case of equality between categories, one can break ties by choosing the class label attached to the closest neighbor. As a variant, one can attribute not the same importance to each of the $k$ neighbors, but a weight inversely proportional to the distance to the query text. The final decision is taken by

[4] In stylometric studies, error in the data could also occur in the label (e.g., the mentioned author was not the true one).

a weighted vote. Of course, defining an efficient $k$ value is not evident, and different solutions must be evaluated to verify their effectiveness.

The $k$-NN method is simple to understand and can provide arguments to justify the final decision. As drawbacks, the classifier is sensitive to the presence of irrelevant or noisy attributes. Thus it is a pertinent idea to apply a feature selection procedure before building a $k$-NN classifier (see Sect. 5.3–5.5). In addition, and similar to the naïve Bayes model (see Sect. 6.2), having many correlated attributes can increase their importance when computing the distance.

The computation cost is also another disadvantage of this method. To identify the $k$-NN one must compute the distance to all text surrogates occurring in the training set (instance-based approach). This weakness could be amended by aggregating all vectors belonging to the same category to generate a unique profile. To classify a query text, it is enough to compute the distance with all profiles and to select the closest one. As a variant, instead of having a single profile per class, the system could represent each category by a reduced set of typical instances [420].

As another variant to reduce the computation cost, the training sample can be organized into a $k$D-tree (or $k$D-ball) to avoid computing the distance with all instances but only for a few of them [420]. Usually, such hidden data structures are implemented by libraries performing the $k$-NN method without modifying the final result.

## 6.2 Naïve Bayes

The *multinomial naïve Bayes model* [269, 420] is a typical text classifier derived from the machine learning paradigm. With this approach, each possible category (or author) is called a hypothesis and denoted as $H_j$ for $j = 1, 2, \ldots r$. To define the most likely category of a query text Q, the naïve Bayes model selects the one maximizing Eq. 6.14, in which $t_{i,Q}$ represents the $i$th term in the sequence of terms appearing in the query text Q, and $n_Q$ indicates the length of this query text. This equation combines two estimations, namely the *prior* (denoted by $p(H_j)$) and the *likelihood* to determine the maximum a posteriori (or MAP).

$$H_{MAP} = \max_{H_j} \; p(H_j|Q) \propto \underbrace{p(H_j)}_{\text{prior}} \cdot \underbrace{\prod_{i=1}^{n_Q} p(t_{i,Q}|H_j)}_{\text{likelihood}} \tag{6.14}$$

$$H_{MAP} = \max_{H_j} \; p(H_j|Q) \propto \; p(H_j) \cdot \prod_{i=1}^{m} p(t_i|H_j)^{tf_{i,j}} \tag{6.15}$$

As an alternative view, the likelihood can be computed as shown in Eq. 6.15 where each $p(t_i|H_j) = p(t_{i,Q}|H_j)$ reflects the same occurrence probability present

in Eq. 6.14 but at the power $tf_{i,j}$. In Eq. 6.14, the multiplication is performed for each term in the word sequence appearing in the query text. For example, if the word *while* appears five times in a text, its occurrence probability will occur five times in Eq. 6.14, but only once, to the fifth power, in Eq. 6.15. Thus, the multiplication is done over all tokens in Eq. 6.14 and only for all distinct word-types (denoted by $m$) in Eq. 6.15.

The *prior* $p(H_j)$ indicates the probability that $H_j$ is the correct category (or author) without considering any further information. When faced with a non-informative prior, the full ignorance can be modeled by a uniform distribution over all categories. Each class has the same probability to be the true one and $p(H_j)$ is estimated as $1/r$ (with $r$ denoting the number of categories). As an alternative, one can take into account the proportion of texts occurring in each category in the corpus. If one can count 500 texts in the training corpus and 100 are belonging to class $S$, then the prior $p(S) = 100/500 = 0.2$. As an alternative, one can make some educated guess or subjective judgment. Thus, each $p(H_j)$ can be individually estimated while respecting that the sum of all probabilities (or all hypotheses) is equal to 1.0 (see Eq. 6.16).

$$\sum_{j=1}^{r} p\left(H_j\right) = 1.0 \tag{6.16}$$

The second part of Eq. 6.15 is called the *likelihood* and estimates the probability that one can observe the term sequence appearing in the query text knowing that the category (or author) is $H_j$. To estimate this probability, the *correct Bayesian model* implies to estimate the following probability:

$$p(t_{1,Q} \wedge t_{2,Q} \wedge t_{2,Q} \wedge \cdots t_{n_Q,Q} | H_j) \tag{6.17}$$

in which one needs to have an *accurate estimation* of the sequence of all the terms $t_{i,Q}$ in Q. This is practically impossible (for a realistic length $n_Q$). One cannot obtain a huge amount of textual data to accurately estimate all possible interactions between terms under the constraint that this text sample belongs to category $H_j$.

To simplify this computation, the formulation in Eq. 6.15 assumes that there is no association between words. The occurrence probability of each term $t_i$ in a document Q is denoted $p(t_i|H_j)$ and it is estimated independently, without considering terms that could occur before or after it, or even the frequency of other features. Of course, this is not fully realistic and corresponds to a *naïve* view. For example, when the word *boat* appears in a text, one can expect to find related terms such as *harbor*, *sea* instead of *electron* or *book*. According to this unsophisticated view, an estimation of the conjunction of all terms is simply a multiplication of each of them.

The remaining problem is to obtain an accurate estimation of the term probabilities $p(t_i|H_j)$, for all terms $t_i$ and for each hypothesis $H_j$. To achieve this, all texts belonging to the same category (or author) are concatenated to form a profile.

For each term $t_i$, the occurrence probability is estimated by the ratio between its occurrence frequency in the profile $H_j$ (denoted $tf_{i,j}$) and the length of this sample ($n_{H_j}$) (see Eq. 3.4).

Let us take a toy-size example. Table 6.3 displays the contents of five hypothetical tweets forming the training set of our corpus. Two categories must be identified, texts written by a male (labeled M) or female (F) author. To estimate the prior associated with both categories, one opts for a non-informative prior ($p(M) = p(F) = 1/2$) or according to the number of tweets in each category ($p(M) = 2/5$, $p(F) = 3/5$).

**Table 6.3** Five tweets written by men (M) and women (F)

| Tweet ID | Text | Category |
|---|---|---|
| 1 | football tv money tv | M |
| 2 | sport tv football sport friends | M |
| 3 | family | F |
| 4 | family friends family | F |
| 5 | friends tv sport | F |
| Q | friends friends tv | ? |

To estimate the occurrence probability of each word in each category, all tweets according to each class are concatenated. For male writers, the resulting text length is 9 and for females 7. To estimate the occurrence probability of the word *tv* knowing that the writer belongs to category M, we compute the ratio between the term frequency in that class ($tf_{i,j} = 3$) and the text length of this category ($n_M = 9$), resulting in an estimation of $3/9$. Table 6.4 shows both the direct estimation for all words and both categories. In addition, the last two columns display the same estimations with Laplace's smoothing (see Sect. 3.2). In this latter case, the numerator is increased by one, and the denominator is added by the number of distinct words in our corpus (namely $m = 6$).

**Table 6.4** Occurrence probability estimations for the category men (M) and women (F)

| | Direct | | Laplace | |
|---|---|---|---|---|
| Word | $p(t_i\|M)$ | $p(t_i\|F)$ | $p(t_i\|M)$ | $p(t_i\|F)$ |
| family | 0/9 | 3/7 | (0+1)/(9+6) | (3+1)/(7+6) |
| friends | 1/9 | 2/7 | (1+1)/(9+6) | (2+1)/(7+6) |
| tv | 3/9 | 1/7 | (3+1)/(9+6) | (1+1)/(7+6) |
| football | 2/9 | 0/7 | (2+1)/(9+6) | (0+1)/(7+6) |
| money | 1/9 | 0/7 | (1+1)/(9+6) | (0+1)/(7+6) |
| sport | 2/9 | 1/7 | (2+1)/(9+6) | (0+1)/(7+6) |

Grounded on estimations shown in Table 6.4, one can compute the likelihood of observing the query tweet Q displayed in the last row of Table 6.3. This disputed

tweet contains three tokens (but two word-types). Using a direct estimation and following Eq. 6.15, the likelihood for category M is provided by Eq. 6.18 and for the other class by Eq. 6.19.

$$p(friends|M)^2 \cdot p(tv|M)^1 = \left(\frac{1}{9}\right)^2 \cdot \frac{3}{9} = 0.0041 \tag{6.18}$$

$$p(friends|F)^2 \cdot p(tv|F)^1 = \left(\frac{2}{7}\right)^2 \cdot \frac{1}{7} = 0.0117 \tag{6.19}$$

The direct estimation assigns a probability of zero to words that never occur in a given category. In our example, this is the case for *family* with category M and *money* and *football* for class F. If a term does not occur in the training set, it is harsh to conclude that this term will never appear. Thus, it is wise to apply a smoothing approach to these probability estimates as explained in Sect. 3.2. With the Laplace's smoothing, the following likelihoods are obtained:

$$p(friends|M)^2 \cdot p(tv|M)^1 = \left(\frac{2}{15}\right)^2 \cdot \frac{4}{15} = 0.0047 \tag{6.20}$$

$$p(friends|F)^2 \cdot p(tv|F)^1 = \left(\frac{3}{13}\right)^2 \cdot \frac{2}{13} = 0.0082 \tag{6.21}$$

Using a uniform distribution for the prior, Eq. 6.22 combines the prior and the likelihood for the male category, and Eq. 6.23 for the female one.

$$\text{Male:} \quad \frac{1}{2} \cdot \left(\left(\frac{2}{15}\right)^2 \cdot \frac{4}{15}\right) = 0.00237 \tag{6.22}$$

$$\text{Female:} \quad \frac{1}{2} \cdot \left(\left(\frac{3}{13}\right)^2 \cdot \frac{2}{13}\right) = 0.0040965 \tag{6.23}$$

These values are not directly the probability estimations but are proportional to the corresponding probabilities. To compute the exact probabilities, Eq. 6.24 indicates that a normalization factor (denominator of Eq. 6.24) must be taken into account.

$$p(H_j|Q) = \frac{p(H_j) \cdot \prod_{i=1}^{m} p(t_i|H_j)^{tf_{i,j}}}{\sum_{k=1}^{r} p(H_k) \cdot \prod_{i=1}^{m} p(t_i|H_k)^{tf_{i,k}}} \tag{6.24}$$

This normalization factor is the summation over all possible categories (to ensure that the sum over all possible outcomes returns 1.0). In our case, the resulting probabilities are provided in Eqs. 6.25 and 6.26, signaling that the chance that the

query tweet was written by a woman is higher than for a man.

$$p(\text{Male}|Q) = \frac{0.00237}{0.00237 + 0.0040965} = 0.3665 \tag{6.25}$$

$$p(\text{Female}|Q) = \frac{0.0040965}{0.00237 + 0.0040965} = 0.6335 \tag{6.26}$$

Even if the probabilities are given in Eqs. 6.25 and 6.26, Eq. 6.15 does not impose that the probabilities must be computed. One can simply select the hypothesis maximizing Eq. 6.15. In our example, the maximum value over Eqs. 6.22 and 6.23 indicates that class F achieves the highest score. When the final decision must be explained to the user, it is wiser to also deliver the probabilities associated with each possible class. These values are easily interpreted and indicate the classifier certainty about the proposed attribution.

As a variant of this multinomial naïve Bayes, one can consider that the frequency or the repetition of a term or attribute is not important. The emphasis must be placed on the *presence* and *absence* of a given stylistic feature according to the underlying category (or writer). As explained in Sect. 5.2, some DNA-based text representation strategies are grounded on a set of binary attributes. In such cases, it makes sense to take account of multiple (multivariate) binary (Bernoulli process) variables and the resulting model is called *multivariate Bernoulli naïve Bayes*.

The prior probability estimations are computed as previously. However, to estimate the likelihood, $p(t_i|H_j)$ is estimated as the ratio between the number of texts in category $H_j$ having, at least, one occurrence of the term $t_i$ and the number of texts belonging in category $H_j$. According to data shown in Table 6.3, the term *family* occurs in two texts written by a woman over a total of three tweets in this class. The probability of the presence of this term is $2/3$. To estimate the probability of the absence of this term in a tweet written by women, one can simply subtract the probability of the presence from one as, in our example, $1 - 2/3 = 1/3$. Table 6.5 reports all the probability estimates for our example.

**Table 6.5** Occurrence probability estimations for the category men (M) and women (F)

| | Direct | | Laplace | |
|---|---|---|---|---|
| Word | $p(t_i\|M)$ | $p(t_i\|F)$ | $p(t_i\|M)$ | $p(t_i\|F)$ |
| family | 0/2 | 2/3 | (0+1)/(2+2) | (2+1)/(3+2) |
| friends | 1/2 | 2/3 | (1+1)/(2+2) | (2+1)/(3+2) |
| tv | 2/2 | 1/3 | (2+1)/(2+2) | (1+1)/(3+2) |
| football | 2/2 | 0/3 | (2+1)/(2+2) | (0+1)/(3+2) |
| money | 1/2 | 0/3 | (1+1)/(2+2) | (0+1)/(3+2) |
| sport | 1/2 | 1/3 | (1+1)/(2+2) | (1+1)/(3+2) |

With the Bernoulli model, the query tweet "friends friends tv" is represented as a sequence of words present or absent as, in our example, "-family, friends, tv, -

football, -money, -sport." If a word occurs more than once (e.g., *friends* in the query text), its repetition is simply ignored.

For category M, the direct estimation is reported in Eq. 6.27, while Eq. 6.28 indicates the estimation for class F. In this computation, the prior is non-informative (1/2 for each gender).

$$\text{M: } \frac{1}{2}\cdot\left(1-\left(\frac{0}{2}\right)\right)\cdot\frac{1}{2}\cdot\frac{2}{2}\cdot\left(1-\left(\frac{2}{2}\right)\right)\cdot\left(1-\left(\frac{1}{2}\right)\right)\cdot\left(1-\left(\frac{1}{2}\right)\right)=0 \tag{6.27}$$

$$\text{F: } \frac{1}{2}\cdot\left(1-\left(\frac{2}{3}\right)\right)\cdot\frac{2}{3}\cdot\frac{1}{3}\cdot\left(1-\left(\frac{0}{3}\right)\right)\cdot\left(1-\left(\frac{0}{3}\right)\right)\cdot\left(1-\left(\frac{1}{3}\right)\right)=0.0247 \tag{6.28}$$

Clearly, the Bernoulli model takes account of both the presence and absence of all word-types appearing in the vocabulary or all stylistic markers. As depicted in Eq. 6.27, the direct estimation can result in a zero probability. The corresponding hypothesis is therefore immediately rejected. A better approach is given in Eqs. 6.29 and 6.30 based on Laplace's smoothing.

$$\text{M: } \frac{1}{2}\cdot\left(1-\left(\frac{1}{4}\right)\right)\cdot\frac{2}{4}\cdot\frac{3}{4}\cdot\left(1-\left(\frac{3}{4}\right)\right)\cdot\left(1-\left(\frac{2}{4}\right)\right)\cdot\left(1-\left(\frac{2}{4}\right)\right)=0.008789 \tag{6.29}$$

$$\text{F: } \frac{1}{2}\cdot\left(1-\left(\frac{3}{5}\right)\right)\cdot\frac{3}{5}\cdot\frac{2}{5}\cdot\left(1-\left(\frac{1}{5}\right)\right)\cdot\left(1-\left(\frac{1}{5}\right)\right)\cdot\left(1-\left(\frac{2}{5}\right)\right)=0.018432 \tag{6.30}$$

After normalization, the two probabilities are reported in Eqs. 6.31 and 6.32 indicating that the author of the query tweet could be a woman.

$$p(\text{Male}|Q)=\frac{0.008789}{0.008789+0.018432}=0.323 \tag{6.31}$$

$$p(\text{Female}|Q)=\frac{0.018432}{0.008789+0.018432}=0.677 \tag{6.32}$$

The Bernoulli model makes sense when the attributes are binary. Moreover, it is one of the learning models that accounts for both the presence and *absence* of a given stylistic feature. However, when the attributes are words with their frequencies, the classical multinomial naïve Bayes tends to provide a better effectiveness. Having noisy attributes is usually not an important problem because they incline to have a similar occurrence probability over all classes. However, having many strongly correlated features will introduce a bias in the final decision. Finally, the effectiveness of the naïve Bayes is usually relatively high and it could be used as a baseline to verify the performance of more complex learning models.

## 6.3 Support Vector Machines (SVMs)

This section discusses the support vector machine (SVM) model, a popular approach in machine learning and considered as an effective approach for solving both the authorship attribution and author profiling problems. As the underlying computation is rather complex compared to other models described in this book, only the main idea will be exposed without describing all the technical details that can be found in [156, 179]. Different implementations are available for different languages (e.g., Python, Java) as well as R libraries (e.g., `e1071`, `libsvm`, `probSVM`), and SVM is also included in the `stylo` package.

Let us start this presentation with the *Federalist Papers* corpus from which a sub-corpus has been extracted. Only two authors are taken into account, namely Hamilton with 45 articles[5] and Madison with 14 texts. The stylistic aspects of these articles are represented by the relative frequencies of only two prepositions *to* and *upon* (computed in ‰). This limitation allows us to visualize all these undisputed papers in two dimensions.

To correctly interpret the following scatterplots, the two axes are orthogonal (or perpendicular) but not orthonormal (different unit vector lengths are used to define both axes). In Fig. 6.2, Hamilton's papers are displayed with a red diamond and each blue circle indicates one Madison's article. As this figure focuses on a fraction of the feature space, few articles are not visible. Hamilton's texts present a high frequency for both prepositions and thus they appear on the top right. In contrast, Madison's texts appear on the bottom left, reflecting low frequencies in the usage of *to* and *upon*.

With the simplest SVM model, also called *support vector classifier*, the learning scheme must define a linear border to separate all instances of both authors (or categories). As our example included two attributes, this class boundary corresponds to a line. When three attributes are present, a plan determines the border, and with more than three features, a hyperplane must be defined.

Figure 6.2 shows this linear class boundary by a solid black line. Clearly, this line separates all articles written by Hamilton from those authored by Madison. When considering the entire set of 51 articles by Hamilton, one cannot define a linear border perfectly separating all instances as depicted in Fig. 6.3. For example, article M38 appears on the incorrect side of the border.

To define this class boundary, SVM does not need all instances (or all text representations) but only a reduced set denoted $S$ containing all *support vectors* (or points). In Fig. 6.2, the border is determined with the text representations labeled H59, M38, and M40. To identify them, a star was superimposed on these three points. In Fig. 6.3, the linear border was identified through the points H25 and M42.

[5]To simplify the beginning of this presentation, six articles authored by Hamilton have been ignored (namely #6, #9, #11, #21, #32, and #84). After removing these texts and as it will be shown later, a clear linear class boundary can be drawn between papers authored by Hamilton to those by Madison.

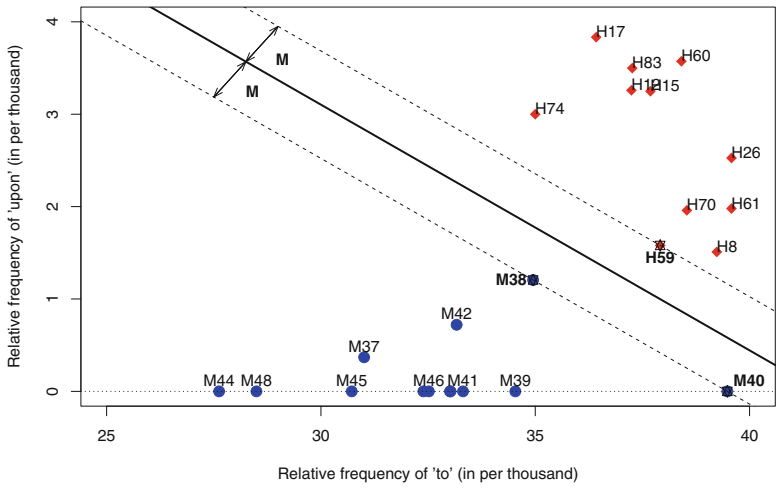


**Fig. 6.2** Our first SVM model with two attributes *to* and *upon*

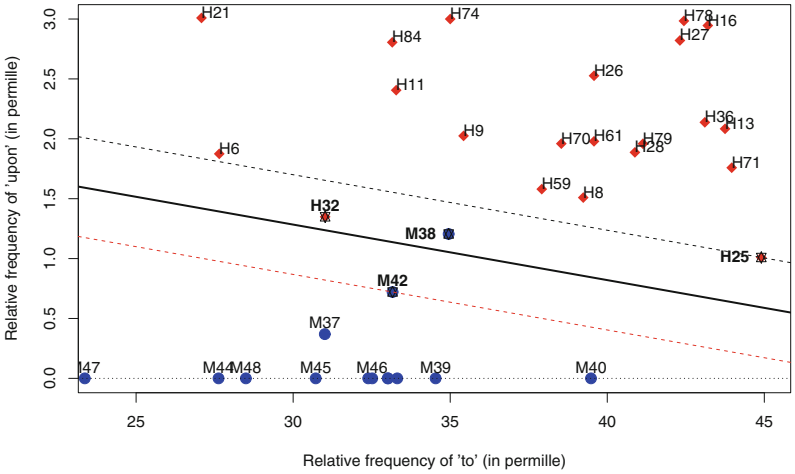


**Fig. 6.3** A second SVM model with two attributes *to* and *upon*

In this scatterplot, a second source of support vectors arises with points occurring inside the margin (e.g., H32) or appearing on the wrong side of the border (e.g., M38). A star is also superimposed to those points.

Based on Fig. 6.2, many lines can be drawn to perfectly split all instances according to their author. Other examples of such linear borders separating the two authors are depicted in Fig. 6.4. Thus, perfectly separating the two classes is not the unique and mandatory criterion. It is more important that the class boundary be as far away as possible from all points to generalize well the distinction between the two styles. This principle is fundamental. To define a good classifier, it is not

mandatory to achieve a low error rate on the *training* sample. The key objective is to learn from the training set the class characteristics to generalize well their intrinsic traits. The essential point is to be able to achieve high performance on new unseen instances (included in the test sample).

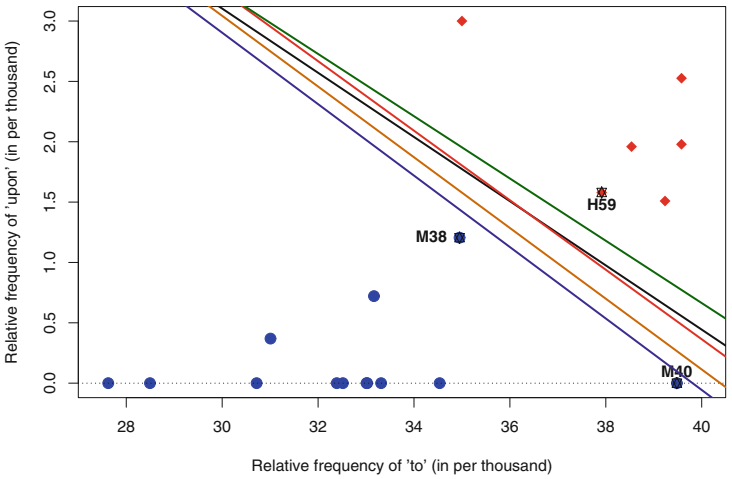


**Fig. 6.4** Multiple examples of linear class boundaries

To formalize this concept, the margin is defined as the distance between the closest point(s) and the class boundary. This distance is indicated in Fig. 6.2 with the label "M." The same distance appears on the left and on the right side. When determining this class boundary, SVM selects the one producing the largest margin or, in other words, the one that generalizes well the differences between the two authors. In Fig. 6.2, three points have a distance M to the border, namely the points H59, M38, and M40.

In Fig. 6.3, the margin is slightly smaller than those presented in Fig. 6.2. The two article surrogates defining the class boundary are H25 and M42 and for them, the distance to the border is equal to M. However, one can see in Fig. 6.3 that two points (H32 and M38) appear inside the margin. Do these two points correspond to classification errors? Not a "real" one for H32 because it is on the correct side of the boundary. The single real error is M38 appearing on the wrong side of the class boundary. How is it possible that the learning scheme produces an error (on the training set)?

When defining this class boundary, the underlying criterion is to maximize the margin and to reduce the risk of returning an overfitted solution, with respect to the training sample. This second goal could be estimated by measuring the variation in the coefficients defining the class boundary (or the values of the slope and intercept when faced with two attributes). If, when adding or removing one (or a few) point(s), large differences in the coefficient values occur, the underlying model

could be viewed as an overfitted one. SVM accepts a non-perfect class separation if the overfitting risk is reduced or when greater robustness is achieved (a model less sensitive to errors in the data). In our example, SVM opts for a larger margin implying generating one error. We must insist that the real criterion is to achieve a high accuracy rate on test instances, not on the training ones.

To allow the learning scheme to produce a better class boundary without perfectly separating the classes, a parameter called *cost* (usually denoted by the variable $c$) controls the number and the amplitude of the errors. This value can be interpreted as a budget that can be spent among the classification errors. A first source of errors are points present inside the margins but on the correct side of the border. For them, the impact on the budget is small and limited. When defining the margin as a unit distance, each point in the margin, but on the correct side, represents a cost less than one [179]. In fact, the real cost corresponds to its distance to the margin. The second error type corresponds to points on the wrong side of the border. Their cost is larger, for example, for point M38 in Fig. 6.3.

After determining the class boundary and S, the set of support vectors, SVM can classify new instances. To assign the disputed text to one of the two authors (or categories), the SVM computes its distance to the border. When this distance is positive, the text is assigned to the first writer, otherwise to the second. For example, Paper Q54 presents a relative frequency of 30.06‰ for the preposition *to* and 1.0‰ for *upon*. As shown in Fig. 6.5, this point belongs to Madison's region. This is a correct assignment. Moreover, all other disputed articles are depicted as green triangles in Fig. 6.5.

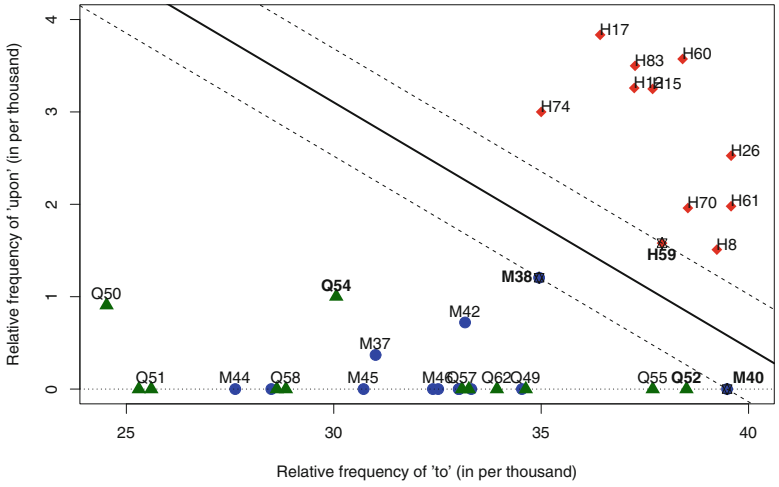


**Fig. 6.5** Our SVM model with two attributes *to* and *upon* and the twelve disputed articles

To numerically verify this attribution, the disputed article Q54 is a vector with its two components [30.06, 1.0]. The projection of this point on the border is performed with a perpendicular line having a $slope = 3.76$ and an $intercept = -112.02$ and

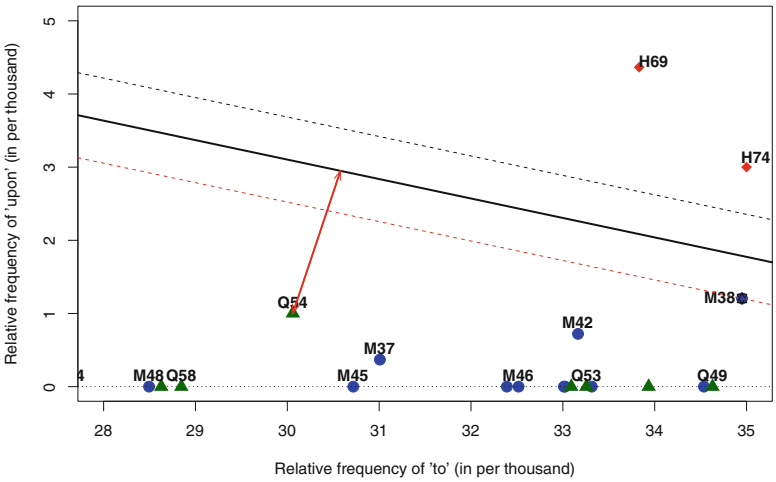


**Fig. 6.6** Detail of the previous figure showing the projection of Point Q54 on the border

shown with a red arrow in Fig. 6.6. The distance to the class boundary is slightly larger than 2.0, a positive value indicating Madison's side. This distance is also larger than the margin indicating a proposed assignment with some certainty. In Fig. 6.5, clearly Q52 occurs very close to the margin but on Madison's side. This attribution is clearly less certain.

To understand the importance of the support vectors, one can view the SVM model in its second form called the dual. In this formulation, the inner product plays an important role. Given two vectors A and B, their inner product, denoted by $\langle A, B\rangle$, is defined by Eq. 6.33 and corresponds to the summation after multiplying each component ($a_i$ and $b_i$, for $i = 1, 2, \ldots, m$).

$$\langle A, B\rangle = \sum_{i}^{m} a_i \cdot b_i \tag{6.33}$$

To classify a new point denoted Q, SVM computes a linear combination of all inner products over all $n$ vectors (denoted $D_i$) belonging to the training sample as indicated by Eq. 6.34. This computation implies a high cost. However, it can be simplified. For all vectors not being a support vector, the coefficient $\alpha_i$ in Eq. 6.34 is 0. Thus, all those points should be ignored. The inner product must be computed only for all support vectors (having a value $\alpha_i > 0$). Determining the set $S$ of support vectors is precisely the task performed during the learning stage. SVM must

determine the value $\alpha_i$ for each of them, as well as the value for the bias $\beta_0$.

$$f(Q) = \beta_0 + \sum_{i}^{n} \alpha_i \cdot \langle Q, D_i \rangle = \beta_0 + \sum_{i}^{|S|} \alpha_i \cdot \langle Q, D_i \rangle \tag{6.34}$$

The number returned by Eq. 6.34 is the distance from the point Q to the class boundary. The sign of this value defines the assigned category. Its magnitude is an indication of the certainty that the classifier has about the proposed decision. A large value specifies that the classifier is relatively certain about the proposed attribution. A small distance signifies a larger uncertainty in the proposed assignment.

The popularity and effectiveness of the SVM classifier are related to its capability to define a non-linear class boundary through the application of kernel functions. The kernel function is a generalization of the inner product and its simplest form is shown in Eq. 6.35 corresponding to a *linear kernel*. As one can see, this first kernel is the classical inner product.

$$K_{linear}(A, B) = \langle A, B \rangle = \sum_{i}^{m} a_i \cdot b_i \tag{6.35}$$

The formulation depicted in Eq. 6.34 could then be replaced by Eq. 6.36. This change does not modify the underlying computation and the class boundary is still linear. To be precise, only when applying a non-linear kernel, the classifier could be called *support vector machines* (SVMs).

$$f(Q) = \beta_0 + \sum_{i}^{|S|} \alpha_i \cdot K_{linear}(Q, D_i) \tag{6.36}$$

Instead of being limited to a linear class boundary, more complex kernel functions can be applied, for example, a polynomial kernel of degree $d$ defined by Eq. 6.37 or a radial kernel in Eq. 6.38.

$$K_{polynomial}(A, B) = \left( \gamma + \sum_{i}^{m} \alpha_i \cdot b_i \right)^d \tag{6.37}$$

$$K_{radial}(A, B) = \exp\left( -\gamma \cdot \sum_{i}^{m} (\alpha_i - b_i)^2 \right) \tag{6.38}$$

Selecting a non-linear kernel function allows the SVM classifier to draw a non-linear class boundary. As an example, Fig. 6.7 shows such a border perfectly separating Madison's and Hamilton's papers when using the relative frequency of *the* and *by*. The class boundary must be non-linear in this context and a polynomial kernel function was applied.

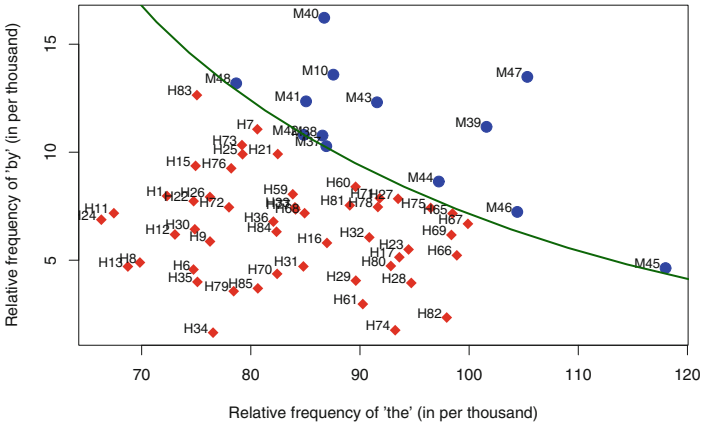


**Fig. 6.7** A non-linear class boundary between Hamilton and Madison's papers

The usefulness of applying a kernel function could also be illustrated with the following example. In Fig. 6.8, one can observe the original sample composed of eight points, four for each class. The first category appears in the center with red triangles, and the second outside with blue circles. In this case, it is not possible to draw a linear border to perfectly split the eight instances according to their two categories.

Following the algebraic development reported in Eq. 6.39, one can see that the original coordinates (e.g., for the point $[a_1, a_2]$) are transformed into a new three-dimensional coordinate system (e.g., $[x = a_1^2, y = \sqrt{2} \cdot a_1 \cdot a_2, z = a_2^2]$). This operation generates a new space with one additional dimension (or attribute). This technique seems an inefficient solution. In Sect. 5.4, different feature selection methods were described to reduce the number of attributes. Why are we now adding features and increasing the feature space?

$$
\begin{aligned}
K_{polynomial}(A, B) = \left( \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \right)^2 &= [a_1 \cdot b_1 + a_2 \cdot b_2]^2 \\
&= a_1^2 \cdot b_1^2 + 2 \cdot a_1 \cdot b_1 \cdot a_2 \cdot b_2 + a_2^2 \cdot b_2^2 \\
&= \begin{bmatrix} a_1^2 \\ \sqrt{2} \cdot a_1 \cdot a_2 \\ a_2^2 \end{bmatrix} \cdot \begin{bmatrix} b_1^2 \\ \sqrt{2} \cdot b_1 \cdot b_2 \\ b_2^2 \end{bmatrix}
\end{aligned}
\tag{6.39}
$$

As one can see in Fig. 6.9, in the augmented new space a linear class boundary can be defined that clearly separates the two categories. In our example, adding one dimension simplifies the determination of a linear border.

This kernel trick is a powerful technique to improve the capability of the SVM model. However, the recurrent problem with the SVM classifier is to determine the

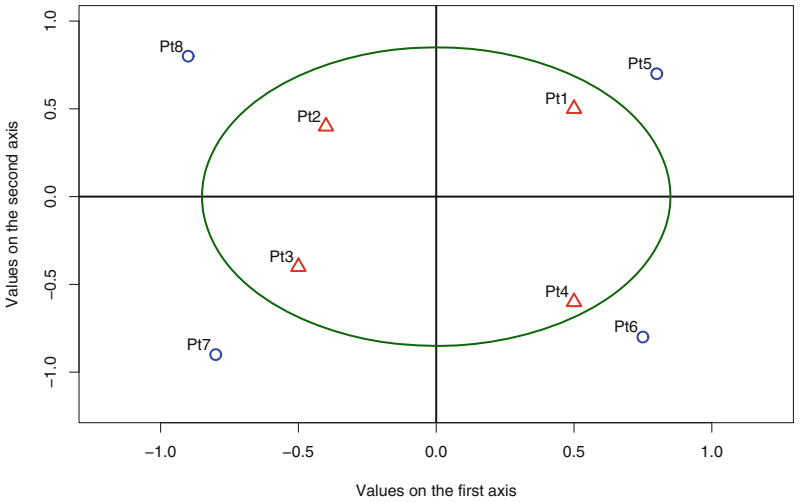


**Fig. 6.8** Original data of a non-linear class boundary between the two classes

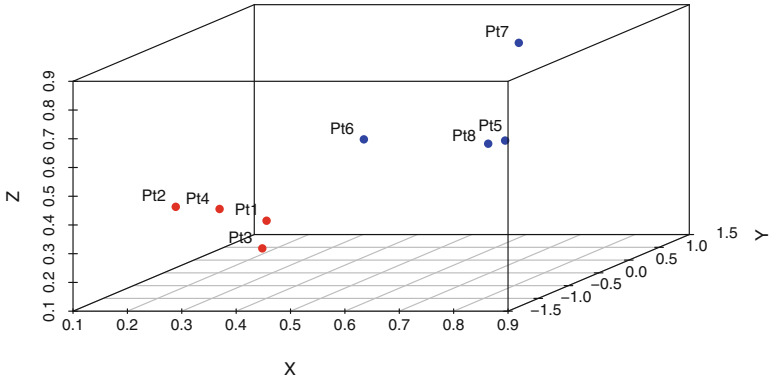


**Fig. 6.9** After applying the kernel function, the original non-linear class boundary between the two classes is now linear

best kernel function and its parameter value(s) (e.g., the degree $d$ for the polynomial function and the value for the cost $c$). No theory explains and clearly justifies which kernel function must be applied with the data in hand. As a starting solution, we recommend to begin with a linear kernel and to adjust for the parameter $c$. Then other kernel functions can be tested and retained if their resulting performance is significantly better than the linear one. In other words, simplicity first.

The SVM model seems limited to draw a class boundary between only two classes. However, a classification problem with more than two categories could be solved with SVM. One way to solve this is to build one SVM per class. Each of these classifiers is generated to discriminate between a single class vs. all the others

(one-versus-all). The final assignment is the one presenting the largest distance to the class boundary.

## 6.4 Logistic Regression

As for the SVM model, the logistic regression is recognized as an effective machine learning model for solving various stylistic or more generally text classification tasks. The term *regression* is well-known in the context of the linear regression, a useful technique to predict a numerical value based on a single (simple linear regression) or a set of independent variables (multiple linear regression). The logistic regression shares some parallels with this familiar model [179].

To illustrate the basic concepts of the logistic regression, let us start with an example related to the automatic classification of tweets. During the CLEF PAN 2019 evaluation campaigns [81], one of the tasks was to identify whether a set of tweets was written by a human being or generated by a bot.[6] If such a recognition is possible with a high accuracy rate, one can have the opportunity to remove spam tweets from the Internet. As the task could be difficult when limited to a single tweet, each identification is based on a set of 100 tweets originating from the same source.

In Sect. 5.2, different text representation techniques have been discussed and some of them do not directly use the words but take account of other attributes. When faced with tweets, one can observe various forms of abbreviations (e.g., `lol`, `btw`, `tyt`), emojis (e.g., 🚑, 🔥), in which some are more related to emotions (e.g., 😂, 😍), as well as mentions (e.g., `@nytimes`, `@SteveCase`), retweets, hashtags (e.g., `#ActOnClimate`), or hyperlinks (e.g., `https://www.t.co/XCJdgOeLlP`).

To discriminate between a set of tweets sent by a human (having the label H) or a bot (B), Table 6.6 presents three examples of both classes. The first column indicates the tweet set identifier, and then the number of tokens (under the label "Size"), the number of emojis (label "Emoji") and from those, the number of emojis symbolizing a face (e.g., 😳, 😘). Under the label "Mention," the number of mentions is reported, while in the next column, the number of hyperlinks is provided ("Link") followed by the number of hashtags ("Hashtag") and retweets ("Retweet"). In the following computation, the examples occurring in Table 6.6 will be completed by 94 other instances forming together a sample of 100 exemplars (this dataset is available in the dedicated webpage). In this sample, 61 have been sent by a bot and 39 by a human being.

As a first analysis, the number of mentions seems to be a pertinent predictor for identifying whether or not a set of tweets was written by a human. This strategy is grounded on the hypothesis that humans will employ more mentions, include more

[6] Chapter 9 presents an extended version on this question.

**Table 6.6** Examples of tweet representations from the bot and human category

| ID | Size | Emoji | Face | Mention | Link | Hashtag | Retweet | Category |
|---|---|---|---|---|---|---|---|---|
| ID-944 | 1168 | 0 | 0 | 0 | 48 | 0 | 0 | B |
| ID-424 | 810 | 0 | 0 | 0 | 100 | 0 | 0 | B |
| ID-233 | 3805 | 0 | 0 | 1 | 202 | 5 | 0 | B |
| ID-2194 | 1611 | 6 | 0 | 21 | 36 | 87 | 5 | H |
| ID-3704 | 2085 | 9 | 5 | 132 | 20 | 21 | 86 | H |
| ID-3293 | 1501 | 33 | 8 | 215 | 30 | 50 | 19 | H |

references to other persons, sources, products, etc. In Table 6.6, this feature presents higher values with humans than with bots. In our example with 100 instances, the mean number of mentions for tweets sent by bots is 12.1, while for humans, this average is 101.54.

To illustrate this classification task using only the number of mentions, Fig. 6.10 displays with a blue square each set of tweets sent by a bot ($y = 0$) and by a red round those sent by a human ($y = 1$). As the number of mentions increases, more tweets written by humans appear. To predict whether or not a set of tweets have been sent by a bot, one can ground the attribution decision on the number of mentions. In this perspective, the higher the number of mentions, the higher the probability that the tweets have been written by a human. In following this idea, the prediction could be based on the linear regression technique. To visualize this proposition, a dark green solid line is displayed in Fig. 6.11.

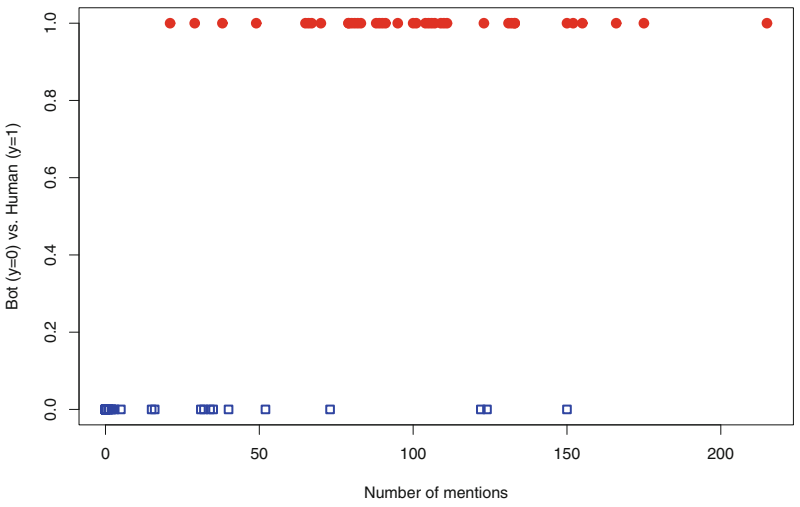


**Fig. 6.10** Set of tweets sent by a bot (0) or a human (1) according to the number of mentions

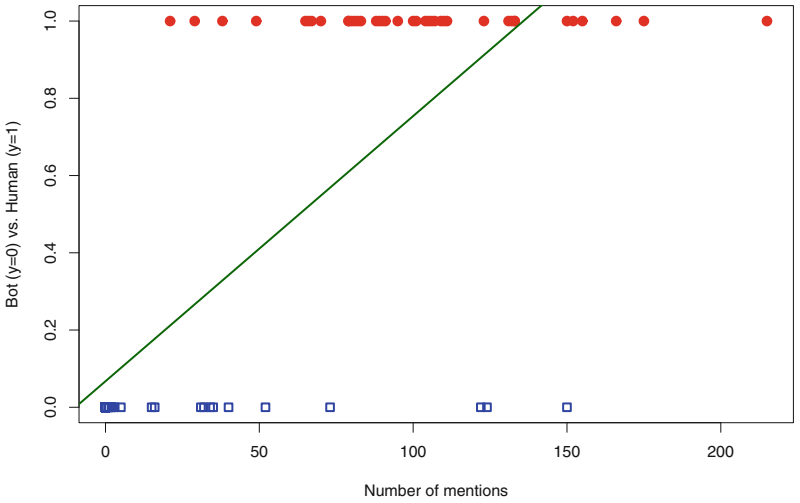


**Fig. 6.11** Linear regression to predict whether a set of tweets was sent by a bot (0) or a human (1) according to the number of mentions

This approach raises some difficulties. After defining the bot category by $y = 0$ and human by $y = 1$, the returned value by the linear regression could differ from these two values, rendering the interpretation problematic. For example, with 100 mentions, the linear regression returns 0.754. Can we interpret this as "corresponding more to category H"? As an alternative, one can view this as the "probability" that the tweets have been written by a human, which is 75.4%. This second interpretation is also tricky. When the number of mentions is larger than 136, the "probability" (computed by the linear regression) is larger than one. Therefore, this solution remains invalid.

To solve these difficulties, one can predict the *probability* that a sample of tweets was written by a human, instead of predicting directly a binary value (0 for a bot, 1 for a human). As the predicted value is a probability, the minimal value must be zero and the maximal 1. Our objective is therefore to generate a function $p(y = 1 \mid tweets)$ returning the probability of having $y = 1$ given a set of tweets. This target function must always return a positive value inside the range $[0 - 1]$.

Different solutions are possible, but as a continuous function the sigmoid defined in Eq. 6.40 fulfills these criteria. In this formulation, the number of mentions is denoted by the variable $l$. As depicted in green in Fig. 6.12, the sigmoid curve matches an S shape.

$$p(y = 1 \mid l) = \frac{\exp(\beta_0 + \beta_1 \cdot l)}{1 + \exp(\beta_0 + \beta_1 \cdot l)} = \frac{1}{(1 + \exp -(\beta_0 + \beta_1 \cdot l))} \tag{6.40}$$

In this model, two unknown parameters ($\beta_0$ and $\beta_1$) must be estimated according to the training sample. With our 100 instances, the R software provides the following

estimations:[7] $\hat{\beta}_0 = -2.954$ and $\hat{\beta}_1 = 0.0506$ (see Sect. 6.5.4). Observing 50 mentions in a set of tweets, the probability that these tweets were written by a human is 0.395. Having 20 mentions instead of 50, the probability is reduced to 0.125.

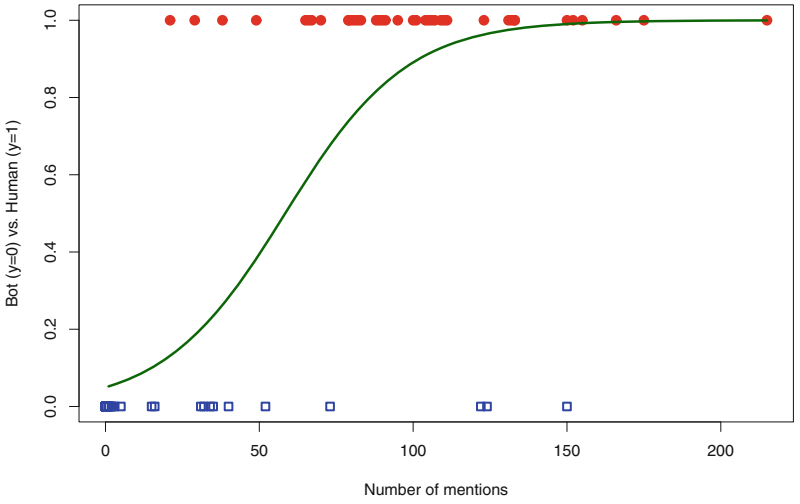


**Fig. 6.12** Example of a logistic regression according to the number of mentions

One of the difficulties with the logistic regression is the estimation of the parameters. To achieve this, we do not have an analytic formula providing a value (e.g., such as $\bar{x}$, the arithmetic average as an estimation of $\mu$, the mean of a population). Therefore, the estimation requires a sequence of iterations that hopefully converges towards a set of values to be assigned to the parameters ($\hat{\beta}_0$ and $\hat{\beta}_1$ in our example). Having only a small sample (e.g., less than 50 instances), these estimations could be difficult and unstable.

When inspecting Table 6.6, one can find other features that could be useful to predict whether or not a set of tweets have been written by a human. For example, the attribute *face* indicates the occurrence number of emojis symbolizing a face. According to values depicted in Table 6.6, one can assume that such emojis signal more of an emotion [355] and thus are more frequent in tweets sent by humans.

Figure 6.13 displays this relationship, showing that many sets of tweets generated by bots have none or very few emojis with a face (blue squares). However, the red dots indicate tweets written by a human where the number of faces is clearly larger. In mean, one can find 8.67 faces in a sample of 100 tweets written by a human vs. 0.312 for the bots. As for the mentions, the higher the number of faces, the higher the probability that the tweets have been sent by a human.

[7] A hat is added to the variable to indicate an estimation and not the real but unknown value.

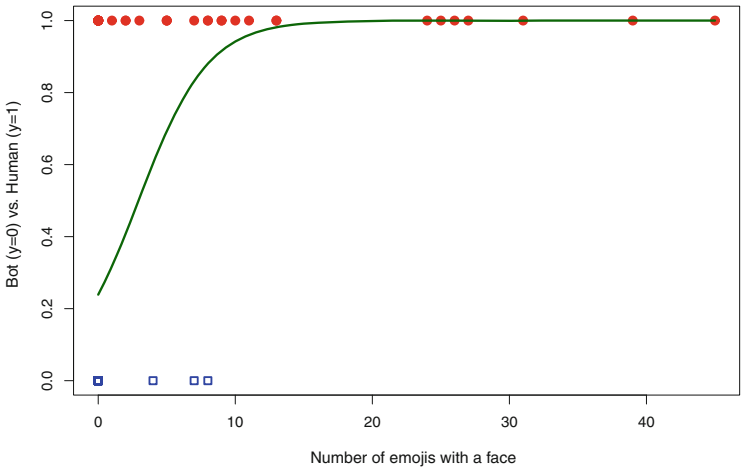


**Fig. 6.13** A logistic regression model with the number of emojis with a face

These first models were rather simple, grounded only on a single predictor. To represent a set of tweets, one can consider combining several attributes as, for example, a subset to those reported in Table 6.6. A more complex logistic regression model can be generated, for example, by taking into account the number of mentions (variable $l$), together with the number of emojis with a face (variable $f$). The resulting logistic regression model is defined in Eq. 6.41.

$$p(y = 1 \mid [l, f]) = \frac{1}{(1 + \exp -(\beta_0 + \beta_1 \cdot l + \beta_2 \cdot f))} \tag{6.41}$$

Using our sample of tweets, one can estimate the values for the three coefficients leading to Eq. 6.42. Clearly, the coefficients associated with the number of mentions and faces have the same sign. As shown previously, a higher number of mentions or faces increases the probability that the tweets have been written by a human.

$$p(y = 1 \mid [l, f]) = \frac{1}{(1 + \exp -(-3.087 + 0.044 \cdot l + 0.165 \cdot f))} \tag{6.42}$$

When in 100 tweets one can count 10 mentions, and 10 emojis with a face, the probability that a human wrote this set is 0.269. Adding 10 mentions and remaining with 10 faces, the probability increases to 0.363.

The logistic regression is not without drawbacks. First, the numerical estimation for defining the parameter values could be unstable, especially when the two classes are well-separated [179, 420, p. 138]. In other cases, the computation could reach the point where the estimated probability is either zero or one, two values corresponding to the asymptotic limits of the logistic function that, from a theoretical point of view, cannot be reached.

Second, it is assumed that, for each feature, the relationship with the binary response is increasing or decreasing. In other words, as the value of the attribute increases, the probability for observing the target category must continuously increase or decrease. As a counter-example, if the two tails of the distribution of an attribute are useful to predict one label but the middle part of the distribution is correlated with the second label, this attribute must be split into two distinct variables.

## 6.5 Examples with R

This section exposes a practical presentation of the previously described models using the R software.[8] We assume that the reader has some basic knowledge of R. Otherwise, a few references are provided in Appendix A.1. The text files with the R code and the needed datasets are available in the dedicated website. In the following examples, the libraries called `class` and `e1071` are needed. They must be downloaded from the CRAN website before trying the proposed R code.

Using our *Federalist Papers* corpus, Sect. 6.5.1 explains how one can propose a solution based on a $k$-NN model based on only six selected words. Section 6.5.2 discusses the naïve Bayes model with a mixed set of predictors, some are numerical, while others are binary. To explain the application of the support vector machines (SVMs) with R, Sect. 6.5.3 continues with the same corpus. Finally, Sect. 6.5.4 exposes the application of the logistic regression model to discover whether a set of tweets have been generated by a bot or written by a human.

### *6.5.1* K-*Nearest Neighbors (*k-*NN)*

To illustrate the practical aspects of the $k$-NN model, the articles of the *Federalist Papers* are represented by the absolute frequency of six selected words, namely *the*, *by*, *upon*, *on*, *would*, and *there*. This information is stored in the file `"FederalistFreq.txt."` With the first instructions depicted below, the content of this file is read and stored in a data frame denoted `mydata`. In this file, the first line signals the variable names (`header=TRUE`) and the first column indicates the instance names (`row.names=1`). The dimensions of this dataset are then printed (`dim(mydata)`). One can find 82 instances with seven variables. As for other files, the same structure is used; the last column always contains the decision. In the next commands, two variables are created to memorize the index of the

[8] The R software can be freely downloaded from the CRAN website (`www.cran.r-project.org/`) together with numerous packages providing functions not present in the basic working space. The examples presented in this section are available in the dedicated webpage.

column containing the decisions (`decIndex`) and the last index for the predictors (`predIndex`). In the current case, one can count six features. To avoid having too many digits after the decimal point, the command `options(digits=3)` limits this number to three. However, this statement is considered as a suggestion by the R system and not as a hard constraint.

With the next command, some instances are displayed with the six terms and the decision. It is always a good practice to view a few instances to quickly verify the correctness of the data. As an alternative, one can enter the `str(mydata)` command.

```
> mydata <- read.table("./Data/FederalistFreq.txt",
          header=TRUE, row.names=1)
> dim(mydata)
   [1] 82 7
> decIndex <- dim(mydata)[2]
> predIndex <- decIndex - 1
> options(digits=3)
> mydata[c(1, 51, 52, 65, 66, 70, 71),]
        the   by  upon  on  would  there  decision
H1      127   14     6   9      2      2         H
H85     240   11    12  17      6     10         H
M10     259   39     0  18      6      6         M
M48     167   28     0  16      3      2         M
J2      105   10     1   8      5      0         J
J64     172   30     0  14      7      6         J
Q49     176   15     0  16     22      2         M
```

Clearly, the instance names (or row names) correspond to the concatenation of the first letter of the author name and the article number (e.g., H85 is the 85th article in this corpus, a text written by Hamilton). Inside this data frame, each attribute is represented by an integer, and the decision by a single letter (H, J, or M). This last component is stored as a factor in R, viewed by the user as a string but the internal representation is an integer, from 1 to the number of different category names (e.g., 3 in the current case).

As one can see, the different absolute frequencies present very different values when considering various attributes. Particularly with the determiner *the*, the amplitude is higher than with other words. Therefore, a normalization must be applied before computing the distance between instances. Three functions are proposed. First, the normalization could be based on the sum of all values (see Eq. 6.1) using the R function `normalizeSum()` or with the MinMax technique shown in Eq. 6.2 using the function `normalizeMinMax()`. As a third approach, one can standardize the values using the function `standardize()`.

```
> normalizeSum <- function(x) {
          return (x / sum(x)) }
```

```
> normalizeMinMax <- function(x) {
          return ((x - min(x)) / (max(x) - min(x))) }
> standardize <- function(x) {
          return ((x - mean(x)) / sd(x)) }
```

This last normalization procedure was applied with the Delta model. Each predictor value is standardized and this operation is performed independently for each column (or each attribute). As shown in the following R example, the standardization produces both positive and negative values. This fact could invalidate some distance measures.

```
# Standardize each column (or attribute)
> mydata.norm <- apply(mydata[,1:predIndex], 2, standardize)
> mydata.norm[1:3,1:5]
        the       by      upon       on          would
H1   -1.0485   -0.492    0.294   -0.271        -1.225
H6   -0.3537   -0.729   -0.140   -0.817        -0.858
H7   -0.0464    0.617    1.379    0.138         3.275
```

Depending on the context and the intent of the analysist, one can normalize by considering the sum over each row or, in other words, according to each paper. In our context, after applying this normalization, each value represents the relative frequency inside each paper (assuming that the paper length corresponds to the sum of all attribute values which is clearly not the case here when considering only six words). For example, the length of paper H1 is 160 tokens. The resulting relative frequency of the determiner *the* is $127/160 = 0.79375$ and for the preposition *by*, this value is $14/160 = 0.0875$. As shown below, the resulting values are unrealistic and still present large differences between the relative frequencies associated with these two features. A closer look reveals that only the determiner *the* depicts large values compared to the others. One possible solution is to include the real paper length instead of estimating it with the sum of the occurrences of only six terms.

```
# Normalize by the sum over each row (or paper)
> mydata.norm <- apply(mydata[,1:predIndex], 1, normalizeSum)
> mydata.norm[1:3,1:5]
          H1       H6       H7       H8          H9
the   0.7937   0.8404   0.6454   0.7488      0.8195
by    0.0875   0.0516   0.0895   0.0531      0.0634
upon  0.0375   0.0188   0.0351   0.0145      0.0195
```

A second solution is to normalize the values per column (or per attribute). In this case, the sum is computed over all frequencies related to the same term assuming that all texts have a similar length (which could be the case in our corpus composed

of newspaper articles). In the following R commands, this is achieved by the second call to the `apply()` function with, as second parameter, the value `2` (along the columns). In this case, the number of *the* in our corpus is 16,849, and for the paper H1, its relative frequency is $127/16{,}849 = 0.00754$.

```
# Normalize by the sum over each column (or feature set)
> mydata.norm <- apply(mydata[,1:predIndex], 2, normalizeSum)
> mydata.norm[1:3,1:5]
        the        by       upon       on          would
H1  0.00754   0.00845   0.0157   0.00999       0.00159
H6  0.01062   0.00664   0.0105   0.00555       0.00477
H7  0.01199   0.01690   0.0289   0.01332       0.04054
```

In the next step, the data frame `mydata.norm` is subdivided into a training and test sample. With the *Federalist Papers*, the first 70 instances[9] correspond to articles with known authorship, while the last 12 rows represent the disputed papers. In addition, two distinct variables memorize the labels associated with the training and test sets.

```
> mydata.train <- mydata.norm[1:70,]
> mydata.train.labels <- mydata[1:70, decIndex]
> mydata.test <- mydata.norm[71:82,]
> mydata.test.labels <- mydata[71:82, decIndex]
```

In a final step, the library `class` containing the `knn()` and its related functions is uploaded. As this function contains a random choice (to break ties), one needs to set the starting point for the random generator (`set.seed(7539)`) to be sure to obtain the same answers as those depicted below.

To apply the $k$-NN model, one needs to call the function `knn()` with, as arguments, the training set (`train=mydata.train`), the test sample (`test =mydata.test`), the labels for the training sample (`cl=mydata.train.labels`), and the number of nearest neighbors to be used for determining the attribution (`k=3`). Inside this function, the Euclidian function is applied to compute the distance between text surrogates. Ties are broken randomly when the same number of neighbors is present for two (or more) categories. The returned vector indicates the assigned category for each test instance. For the first nine positions in this vector, one paper is attributed to Hamilton, three to Jay, and five to Madison. When comparing these suggested attributions with the right labels, only eight attributions are correct (our last instruction).

```
> library(class)
> set.seed(7539)
```

[9] 51 have been written by Hamilton, 14 by Madison, and 5 by Jay.

```
> knn.pred <- knn(train=mydata.train, test=mydata.test,
                  cl=mydata.train.labels, k=3)
> knn.pred[1:9]
   [1] H J M M J M M J M
   Levels: H J M
> sum(knn.pred == mydata.test.labels)
   [1] 8
```

From this first solution, different variants and sensitivity analyses could be performed. First the number of predictors could be limited, for example, by taking into account only three of them, such as *by*, *upon*, and *would*. With the R commands given below, the resulting performance is the same with eight correct assignments. As a second way, the possible authors could be limited to Hamilton and Madison. In this case, the effectiveness is enhanced to ten correct attributions.

```
> aRange <- c(2, 3, 5) # Selecting only three predictors
> knn.pred <- knn(train=mydata.train[,aRange],
          test=mydata.test[,aRange],
          cl=mydata.train.labels, k=3)
> sum(knn.pred == mydata.test.labels)
   [1] 8
> knn.pred <- knn(train = mydata.train[1:65, aRange],
           test=mydata.test[,aRange],
           cl=mydata.train.labels[1:65], k=3)
> sum(knn.pred == mydata.test.labels)
   [1] 10
> summary(knn.pred)
   H  J  M
   2  0  v10
```

### *6.5.2 Naïve Bayes*

To illustrate an application with the naïve Bayes model, we continue with the *Federalist Papers* corpus. To simplify the presentation, the group of possible authors will be limited to Hamilton and Madison. Second, the representation of each newspaper article corresponds to a mix of absolute occurrence frequencies and some Boolean predictors. To be precise, the two prepositions *by* and *upon* are characterized by their occurrence frequency. As the frequencies for the terms *while*, *whilst*, *fully*, and *kind* are rather low per article, the essential aspect is knowing whether or not such terms appear in a text. Therefore, four Boolean variables representing the four words *while*, *whilst*, *fully*, and *kind* complete each text surrogate.

Similar to the *k*-NN application, the first R commands read the file in which the article representations are stored (`FederalistFreqMixed.txt`). The first line indicates the variable names (`header=T`) and the first column the different text

identifiers (`row.names=1`). Two variables are created to memorize the column index with the decisions (`decIndex`) and the number of attributes (`predIndex`).

The last command allows us to show some instances with the six attributes and the corresponding decision. As the term *while* is a reserved word in R, the variable name is modified into `while.` to avoid ambiguity. Inside the file, the different Boolean variables are indicated by a single letter (`T` or `F`) that are synonyms of `TRUE` and `FALSE`. For example, the article H85 contains 11 occurrences of *by*, 12 of *upon*, and, at least, one of *fully*.

```
> mydata <- read.table("./Data/FederalistFreqMixed.txt",
          header=T, row.names=1)
> dim(mydata)
   [1] 77 7
> decIndex <- dim(mydata)[2]
> predIndex <- decIndex - 1
> mydata[c(1, 51, 52, 65, 66, 77), c(1:decIndex)]
         by   upon   while.   whilst   fully    kind    decision
 H1      14      6     FALSE    FALSE    FALSE   FALSE       H
 H85     11     12     FALSE    FALSE     TRUE   FALSE       H
 M10     39      0     FALSE    FALSE    FALSE   FALSE       M
 M48     28      0     FALSE    FALSE     TRUE   FALSE       M
 Q49     15      0     FALSE     TRUE    FALSE   FALSE       M
 Q63     51      0     FALSE     TRUE    FALSE   FALSE       M
```

One of the main advantages of R resides in its capability to visualize the data to help the user to detect patterns and regularities. As an example, the following lines draw a scatterplot with the absolute frequencies of the prepositions *by* and *upon*. First, two variables (`aVar1` and `aVar2`) specify the two selected attributes. The function `plot()` is called with several parameters. The first two indicate the coordinates for the first 65 instances corresponding to the articles with known authorship. Then the range of values for both axes are provided (`xlim` and `ylim`), followed by the appropriate labels (`xlab` and `ylab`).

As it is important to discriminate between texts written by Hamilton and those authored by Madison, we decide to indicate Hamilton's paper with a round (`pch=19`) and those by Madison with a square (`pch=15`). Similarly, the color (`col=`) associated with Hamilton is `red` and `blue` for Madison. To achieve this effect, the decision stored as a factor is used (`mydata[1:65,decIndex]`). This factor variable is viewed as the letter "H" or "M." But internally a factor is stored as integers, with the smallest string in lexicographic order having the value 1, the second the value 2, etc. In our case, "H" appears first and thus has the value 1, while "M" is replaced by 2.

```
> aVar1 <- 1; aVar2 <- 2
> plot(mydata[1:65, aVar1], mydata[1:65, aVar2],
          xlim = c(0, 80), ylim = c(-2, 20),
          xlab = "Frequency of the term 'by' (per thousand)",
```

```
          ylab = "Frequency of the term 'upon' (per thousand)",
          pch = c(19, 15)[mydata[1:65,decIndex]],
          col = c("red", "blue")[mydata[1:65,decIndex]])
> points(mydata[66:77,aVar1], mydata[66:77,aVar2],
          pch = 17, col="black")
> somePoints <- c(66, 70, 71, 77)
> text(mydata[somePoints,aVar1],mydata[somePoints,aVar2],
          adj=c(1,1.5), cex=0.9, col="black"
          labels=row.names(mydata)[somePoints])
```

With the function `points()`, 12 black (`col=`) triangles (`pch=17`) representing the 12 disputed articles are added to the scatterplot. For some of them (defined by the variable `somePoints`), the article identifier is also printed in black, a little bit on the left and lower (`adj=c(1,1.5)`) of the associated triangle. The font size for this label must be 90% of the normal size (`cex=0.9`). The resulting plot is depicted in Fig. 6.14.

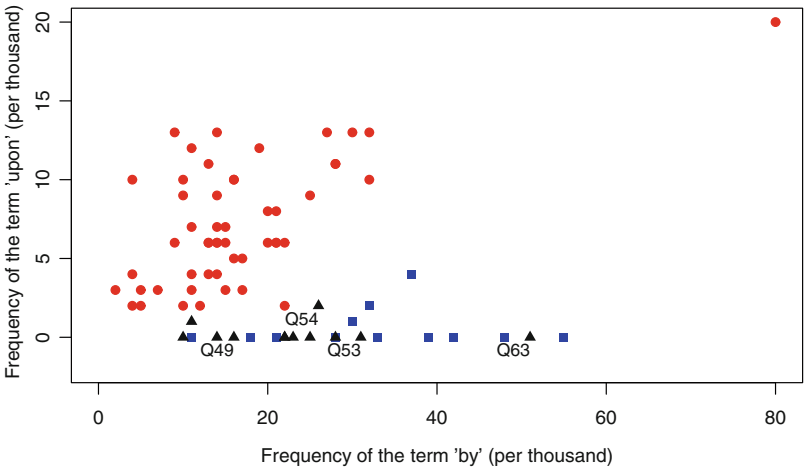


**Fig. 6.14** Example of a scatterplot with the absolute frequency of *by* and *upon*

Having our data frame in the main memory, the text surrogates are split into a training and test sample (the twelve disputed articles). The correct labels for these two sets are stored in two additional variables. Then the corresponding library is loaded (`e1071`).

The function `naiveBayes()` generates the classifier described by the expression `"decision~."` specifying that the binary factor decision must be explained (~) by all other attributes (`.`) present in the associated data frame (or matrix). The second parameter specifies this dataset (`data=mydata.train`).

The classifier is then used to predict the author of the twelve disputed articles with the function `predict()` having as arguments the generated model (`NB.mod1`) and the data frame storing the test sample (`mydata.test`). As depicted below, the

resulting variable (`pred1`) is a factor with the letters "M" and "H." When comparing the proposed attributions and the true ones (`mydata.test.labels`), one can count ten correct assignments. With the function `table()`, the distinction between the two possible sources of errors is presented in a contingency table (see Sect. 4.4). In our case, two texts authored by Madison are attributed incorrectly to Hamilton.

```
> mydata.train <- mydata[1:65,]
> mydata.train.labels <- mydata[1:65, decIndex]
> mydata.test <- mydata[66:77, 1:predIndex]
> mydata.test.labels <- mydata[66:77, decIndex]
> library(e1071)
> options(digits=3)
> NB.mod1 <- naiveBayes(mydata.train$decision~.,
                        data=mydata.train)
> pred1 <- predict(NB.mod1, mydata.test)
> pred1
   [1] M H M M M M H M M M M M
   Levels: H M
> sum(pred1 == mydata.test.labels)
   [1] 10
> table(pred1, mydata.test.labels)
    mydata.test.labels
   pred1 H   M
   H     0   2
   M     0  10
```

With this first model, the probability estimates are based on the maximum likelihood principle or simply by the ratio between the number of successes divided by the number of trials. However, this technique assigns a zero probability to events never seen in the training set which could be viewed as an extreme proposition. It is suggested to apply a smoothing approach to reduce the probability of existing events to the benefit of unseen ones (see discussion in Sect. 3.2). As a simple solution, the Laplace's smoothing (Eq. 3.5 or examples in Table 6.5) could be applied by adding the parameter `laplace=1` when generating the naïve Bayes model. With this strategy, the number of correct attributions is increased by one as depicted in the R statements shown below.

```
> NB.mod2 <- naiveBayes(decision~., data=mydata.train,
                        laplace=1)
> pred2 <- predict(NB.mod2, mydata.test)
> sum(pred2 == mydata.test.labels)
   [1] 11
```

Finally, one can inspect the internal representation of the classifier by simply writing the classifier variable name (`NB.mod1`). Several lines are then printed, starting with the model name and its call. As more pertinent information, one can see the estimated prior probabilities. As the training sample contains 65 articles, and knowing that Hamilton is the author of 51 of them, the prior estimation for this author is simply $51/65 = 0.785$.

The two predictors *by* and *upon* are represented by their absolute occurrence frequencies. Faced with numerical variables, the naïve Bayes model indicates the mean and standard deviation for the two possible categories (Hamilton and Madison in our case). For example, under the label *by*, the mean is 16.6 in texts written by Hamilton and the standard deviation is 11.6. As Madison uses this preposition more frequently, its mean is 32.3 and the standard deviation 11.5. These values are not directly the conditional probabilities but the parameters for estimating them according to, for example, a Gaussian distribution.

```
> NB.mod1
 Naive Bayes Classifier for Discrete Predictors
 Call:
 naiveBayes.default(x = X, y = Y, laplace = laplace)
 A-priori probabilities:
         H     M
      0.785 0.215
Conditional probabilities:
          by
 Y   [,1]   [,2]
 H   16.6   11.6
 M   32.3   11.5
                              upon
 Y   [,1]   [,2]
 H   7.25   3.94
 M   0.50   1.16
                              while.
 Y   FALSE    TRUE
 H   0.608    0.392
 M   1.000    0.000
                              whilst
 Y   FALSE      TRUE
 H   0.9804    0.0196
 M   0.4286    0.5714
                              fully
 Y   FALSE    TRUE
 H   0.745    0.255
 M   0.214    0.786
                              kind
 Y   FALSE      TRUE
 H   0.3137    0.6863
 M   0.9286    0.0714
```

With the Boolean variables, the real conditional probabilities are depicted. For example, the term *while* occurs in 20 articles, all written by Hamilton. Thus, the conditional probability that *while* appears in a text knowing that this article was written by Hamilton is denoted by $p(while = T \mid Hamilton) = 20/51 = 0.392$. Of course, the opposite is $p(while = F \mid Hamilton) = 31/51 = 0.608$ or simply $1 - (20/51) = 1 - 0.392 = 0.608$. As Madison never employs the preposition *while*, the conditional probability that *while* occurs in a text authored by Madison is $p(while = T \mid Madison) = 0/14 = 0$. These probabilities are reported under the label `"while."` in the classifier's description.

### 6.5.3 Support Vector Machines (SVMs)

To present an application based on the SVM classifier, the *Federalist Papers* corpus has been selected with 51 texts authored by Hamilton and 14 by Madison. As predictors, only three terms have been selected, namely *to*, *upon*, and *would*. This reduced set of attributes has been proposed by Fung [124]. The value associated with each feature is the relative frequency in per thousand (or ‰ ) in the underlying newspaper article.

To avoid presenting each value with a long sequence of digits, the presentation of numbers is limited to four digits (`options(digits=4)`). As with the previous examples, the file is read to generate a data frame called `mydata`. The index of the decision variable is memorized (`decIndex`) as well as the number of attributes (`predIndex`). In addition, few examples are provided and the index range corresponding to the training set (`aRange`) or to the test set (`aTestRange`) has been specified.

```
> options(digits=4)
> mydata <- read.table("./Data/FederalistFung.txt",
       header=TRUE, row.names=1)
> dim(mydata)
   [1] 77 4
decIndex <- dim(mydata)[2]
predIndex <- decIndex - 1
> mydata[c(1, 51, 52, 65, 66, 77), c(1:decIndex)]
            to     upon    would        decision
    H1    44.1    3.78     1.26            H
    H85   42.5    4.51     2.26            H
    M10   33.0   0.000     2.00            M
    M48   28.5   0.000     1.61            M
    Q49   34.6   0.000    13.40            M
    Q63   28.6   0.000     3.62            M
> aRange <- 1:65
> aTestRange <- 66:77
```

As it is always pertinent to visualize the data according to the two categories, the following R code produces a scatterplot similar to Fig. 6.2 (to achieve exactly the same graph, six articles written by Hamilton must be removed, namely H6, H9, H11, H21, H32, and H84).

```
> aVar1 <-1; aVar2 <- 2
> plot(mydata[aRange, aVar1], mydata[aRange, aVar2],
       xlab = "Relative frequency of 'to' (in per thousand)",
       ylab = "Relative frequency of 'upon' (in per thousand)",
       col = c("red","blue")[mydata[aRange,decIndex]],
       pch = c(18, 19)[mydata[aRange,decIndex]])
```

In the next R commands, the library (`e1071`) is uploaded. As for the previous machine learning models, the SVM classifier is generated with the expression `"decision~."` In this case, the explaining variables are the relative frequency of *up*, *upon*, and *would*. The data frame corresponding to the training sample (`data=mydata[aRange,]`) appears as the second argument. As additional parameters, a linear kernel is applied (`kernel="linear"`) with a budget of 10 (`cost=10`). As the values of the three predictors are comparable, we do not need to rescale the variables (`scale=F`). By default, the parameter scale is `TRUE`, implying that each variable is rescaled to zero mean and unit variance (or standardized) (see Eq. 6.3).

The command `summary()` presents an overview of the generated model by printing the call and specifying the parameter values. In addition, one can find the number of support vectors (4 in our case), with three in the first class (Hamilton) and one for the second (Madison).

```
> library(e1071)
> svmfit = svm(decision~., data=mydata[aRange,],
          kernel="linear", cost=10, scale=F)
> summary(svmfit)
   Call:
   svm(formula=decision~., data=mydata[aRange, ], kernel=
          "linear", cost=10, scale=F)
   Parameters:
     SVM-Type: C-classification
    SVM-Kernel: linear
       cost: 10
      gamma: 0.3333333
   Number of Support Vectors: 4
    ( 3 1 )
   Number of Classes: 2
   Levels:
    H M
```

The index of the support vectors is stored in the variable `svmfit$index`. With this information, one can identify the corresponding papers

(`mydata[svmfit$index,]`). In our case, one can find three articles authored by Hamilton (H6, H25, and H68) and one (M38) by Madison.

```
> svmfit$index
   [1] 2 16 34 55
> mydata[svmfit$index,]
           to    upon   would   decision
    H6    27.6   1.87    2.81        H
    H25   44.9   1.01   10.60        H
    H68   49.9   1.33    4.65        H
    M38   35.0   1.21    4.52        M
```

After generating our SVM classifier, the twelve disputed articles could be classified. To achieve this, the function `predict()` is called with, as arguments, the classifier (`svmfit`) and the data frame corresponding to the test sample (`mydata[aTestRange,]`). Clearly, the twelve papers are attributed to Madison, a fully correct assignment.

```
> pred <- predict(svmfit, mydata[aTestRange,])
> pred
      Q49    Q50   Q51   Q52   Q53   Q54   Q55   Q56   Q57   Q58   Q62   Q63
       M      M     M     M     M     M     M     M     M     M     M     M
Levels: H M
> sum(pred == mydata[aTestRange,decIndex])
   [1] 12
```

As a second way to generate a test data, a new data frame is created as shown in the following R code. In this example, the three articles correspond to the first three occurring in the test sample. Of course, the proposed attributions, always to Madison, are exact.

```
> newData <-data.frame(to=c(34.6, 24.5, 25.6),
                       upon=c(0.0, 0.908, 0.0),
                       would=c(13.4, 9.99, 4.7))
> pred <- predict(svmfit, newData)
> pred
   1  2  3
   M  M  M
   Levels: H M
```

### 6.5.4 Logistic Regression

We now return to the problem presented in Sect. 6.4. In this example, we need to identify whether a set of tweets was written by a human or generated by a bot. Each set of 100 tweets forms an instance and 100 of them are stored in the file "ExampleBotHuman.txt." Each line represents one observation with seven attributes. In the R code given below, the file is read and the first line stores the variable names (header = T), while the first column (row.names=1) is used as an identifier for each observation.

In a second instruction, the number of observations (100) and attributes (7) included in the data frame mydata is provided. The last column is associated with the decision (bot or human). This last column number is kept in the variable decIndex. Then four randomly chosen instances (#1, #51, #63, and #80) are displayed. The first observation is identified by the string ID-944. This set of tweets contains 1168 tokens, no emojis, no emojis with a face, no mentions, 48 hyperlinks, no hashtags, and no retweets. These tweets have been generated by a bot (decision = B).

```
> mydata <- read.table("./Data/ExampleBotHuman.txt", header=T,
          row.names=1)
> dim(mydata)
   [1] 100 8
> decIndex <- dim(mydata)[2]
> mydata[c(1, 51, 63, 80), 1:decIndex]
           size   emoji   face   mention   link   hashtag   retw   dec
 ID-944    1168      0       0        0       48        0        0      B
 ID-1757   1610      0       0       16       0         1       16      B
 ID-2414   1456      0       0       29       54       31       11      H
 ID-3221   1817     21      10       95       40        2       89      H
> aVar1 <- 4 # The mention variable is in the 4th position
> yValue <- as.integer(mydata[,decIndex]) - 1
> plot(mydata[,aVar1], yValue,
          xlab="Number of mentions",
          ylab="Bot vs. Human",
          col=c("blue","red")[mydata[,decIndex]], lwd=2,
          pch=c(22, 19)[mydata[,decIndex]])
```

In the next instruction, the single predictor to be used for our study is selected. The fourth corresponds to the number of mentions. With the plot() function, a figure is created (corresponding to Fig. 6.10). The independent variable is the number of mentions, and the dependent one is the category label (B or H). Inside the system, this variable is stored with integers, one for each possible value (e.g., 1 for B or 2 for H in our case).

These two values are then used to discriminate between instances belonging to the bot (1) or human (2) category. To discriminate with two colors, the instances with the value "1" will extract the color `blue` and observations with the label "H" the color `red`. The same technique is used to represent the bot instances with the symbol #22 (a square) and the human ones with the symbol #19 (a round).

In the next stage, the simple logistic regression model is built. After limiting the number of digits to three, the call to the function `glm()` generates the logistic model (glm means generalized linear models). As first parameter, the model is described as `dec~mention` indicating that the binary factor `dec` must be explained (~) by the single predictor `mention`. As the second parameter, the sample of observations can be found in the data frame `mydata` (`data=mydata`). The third parameter (`family = binomial`) signals that the target model must be generated according to a logistic regression. The model is stored in the variable `logReg.mod1`. The next instruction asks R to display the important information of this model related to the coefficients.

```
> options(digits=3)
> logReg.mod1 <- glm(dec~mention, data=mydata,
                         family=binomial)
> summary(logReg.mod1)$coef
                 Estimate   Std. Error   z value   Pr(>|z|)
   (Intercept)    -2.9538     0.56014     -5.27    1.34e-07
   mention         0.0506     0.00885      5.71    1.13e-08
```

Under the column "Estimate," one can find the coefficient values ($\hat{\beta}_0$ and $\hat{\beta}_1$) that can be introduced in Eq. 6.40 to derive the estimated model. The next column provides the values for the standard errors (the standard deviation associated with each estimation). The values under the column "z value" are simply the estimate value divided by the standard error[10] (e.g., $0.0506/0.00885 = 5.71$). The corresponding $p$-value is indicated in the last column. This number indicates the probability of obtaining the estimation for the current parameter as it is or more extreme, under the assumption that the real value is zero (in other words, assuming that $H_0$ hypothesis is true). For example, with the second row ($\hat{\beta}_1$), the $p$-value is $1.13e-08$ (a very small positive value). If $H_0$ is true and therefore the real value for $\beta_1$ is 0, the probability of observing a value of 0.0506 or larger is 1.13e−08 (or 0%). Therefore, one can conclude that the number of mentions is really pertinent for this model. The real value for the parameter $\hat{\beta}_1$ is significantly different from 0. Therefore, the number of mentions can be used to predict whether a set of tweets have been generated by a bot or written by a human.

The information provided by the $p$-value is useful to select the pertinent attributes and to remove irrelevant ones. However, one cannot simply build a model with all

[10] Assuming that the null hypothesis $H_0$ is true, meaning that the real value for this estimator is zero.

possible variables and then remove all attributes depicting a $p$-value larger than let us say 5% with the aim of generating the best model. The feature selection must be performed step by step, one variable at a time as discussed in Sect. 5.5.

Based on this model, one can compute the probability that a set of tweets containing 50 mentions was written by a human as follows. In the variable `coefs`, the two estimations are stored. Then one can use them to evaluate the probability computed by the sigmoid function which is 39.5% in our case.

```
> coefs <- logReg.mod1$coefficients
> m <- 50
> 1.0 / (1 + exp(-(coefs[1] + coefs[2]*m)) )
   (Intercept)
     0.395
```

Other logistic models can be studied, for example, adding an explanatory variable to our first model. As a second predictor, one can select the number of emojis symbolizing a face. With these two predictors, a new model is built according to the R code depicted below. Inspecting the $p$-value associated with each attribute, the estimations associated with the first two are significantly different from zero. For the variable `face`, the $p$-value is larger than expected, indicating that this parameter could have a value equal to zero.

```
> logReg.mod3 <- glm(dec~mention+face, family=binomial,
          data = mydata)
> summary(logReg.mod3)$coef
                 Estimate   Std. Error   z value   Pr(>|z|)
   (Intercept)   -3.0871     0.59191     -5.22     1.83e-07
   mention        0.0435     0.00887      4.91     9.27e-07
   face           0.1654     0.10202      1.62     1.05e-01
> newData <- data.frame(mention=c(10, 20, 30),
                        face=c(10, 10, 10))
> newData
        mention   face
      1    10       10
      2    20       10
      3    30       10
> pred <- predict(logReg.mod3, newData, type="response")
> pred
       1      2      3
   0.269  0.363  0.468
```

To predict the label associated with three new observations, a data frame denoted `newData` is generated. In these examples, the number of mentions varies from 10 to 30 but the number of faces is fixed at 10. To determine the corresponding label, the function `predict()` is called always having as the first parameter the

model (`logReg.mod3`) and as the second the data frame containing the new data (or the variable `newData`). With the logistic regression model, the third parameter is `type=response` indicating that the probability estimate is required for each prediction. This probability corresponds to $p(y = 1 \mid an\ observation)$. In our example, the probability that the first set of tweets have been written by a human ($y = 1$) is 0.269. By rising the number of mentions from 10 to 30, this probability increases from 0.269 to 0.468.

To obtain a fair estimation of the success rate, a set of new observations stored in the file denoted `ExampleBotHuman.test.txt` is available. These instances have not been used in the training stage to estimate the coefficients (see explanations in Sect. 4.7). This file comprises 30 instances, 15 have been generated by bots and 15 by humans. To determine the label associated with each new instance, the function `predict()` is again called with as second argument the data frame of the test data (or the variable `mydata.test`). The probability is then estimated and each value larger than 0.5 corresponds to a set of tweets written by a human. In this example, the number of correct attributions is 27 or the success rate is $\hat{p} = \frac{27}{30} = 0.9$.

```
> mydata.test <- read.table("./Data/ExampleBotHuman.test.txt",
          header=T, row.names=1))
> dim(mydata.test)
   [1] 30 8
> pred <- predict(logReg.mod3, mydata.test, type="response")
> anIndex <- pred>0.5
> pred[anIndex] <- 'H'
> pred[!anIndex] <- 'B'
> sum(pred==mydata.test$dec)/length(pred)
   0.9
```

From this success rate estimation, one can build a confidence interval as described in Sect. 4.5. But as the number of instances in the test set is rather limited (30) and as demonstrated by Brown et al. [43], the estimation must be modified according to Eq. 4.16 leading to $\hat{p} = \frac{27+2}{30+4} = 0.853$. The standard deviation of $\hat{p}$ (see Eq. 4.15) is then $0.853 \cdot (1 - 0.853)/\sqrt{30} = 0.0229$. To achieve a coverage of 95%, the confidence interval is then $0.853 \pm 1.96 \cdot 0.0229 = [0.8081; 0.8978]$.

# Chapter 7
# Advanced Models for Stylometric Applications


This chapter presents some methods and approaches proposed to solve the authorship attribution problem and some of them can also be applied for other stylistic questions such as author profiling, author verification, or stylistic change over time (or stylochronometry [50, 378]). Due to the large number of advanced methods suggested during the last two decades, it is not possible to describe all of them. Therefore, this chapter presents only a subset of approaches proposed in the digital humanities domain, as well as some appearing recently in computer science.

The rest of this chapter is organized as follows: Section 7.1 exposes the Zeta method, a specific approach for solving the attribution problem. Section 7.2 explains the application of different compression methods for identifying the true author of a text, while Sect. 7.3 briefly exposes the application of LDA for solving this question. The verification question with some original solutions is discussed in Sect. 7.4. Section 7.5 describes approaches for detecting collaborative work where two or more authors have jointly written a text. Section 7.6 presents the main concepts of neural networks with two applications in authorship attribution. The use of word embedding models to solve the authorship question is presented in Sect. 7.7, while Sect. 7.8 exposes the long short-term memory (LSTM), a recurrent neural architecture derived for the deep learning domain to resolve text categorization tasks implying sequences. The last section exposes the reverse problem or the obfuscation question, or how to develop methods to remove or hide the stylistic traits of an author to circumvent different authorship analyses.

## 7.1 Zeta Method

Within this approach, Burrows [47] proposes to consider words or terms being used recurrently by one author (or category) and rarely or even ignored by another writer. This strategy does not focus on very frequent words but should ground



J. Savoy, *Machine Learning Methods for Stylometry*,
https://doi.org/10.1007/978-3-030-53360-1_7

the attribution on middle frequency terms or "truly idiosyncratic features" said Burrows [47]. Such word-types occur often enough to obtain meaningful counts with the first writer, but not too frequently as to make them undistinctive for the second author. The idea is to contrast texts authored by the first writer (denoted the base set) and, in a counter set, texts reflecting the other one. As indicated, the Zeta test[1] is really appropriate when faced with only two possible outcomes. The counter set, however, could regroup more than one author. But when there are three or more candidates, it is more difficult to have a clear interpretation with this method [320].

Instead of considering each entire text as a distinct instance, each document is divided into non-overlapping chunks or segments of a fixed size $n$, with $n$ ranging from 900 to 6000. In another implementation, one can concatenate all texts of a given category (e.g., written by teenagers) to generate a long document that can then be subdivided. In our example, the base set regroups all articles written by Madison, while the counter set includes newspaper articles written by both Hamilton and Jay. The underlying hypothesis is then to verify whether the disputed articles have been written by Madison or not. When specifying as counter set only articles written by Hamilton, the question is to know whether Madison or Hamilton is the true author of the disputed articles.

The computation of the Zeta value is described in [65, 320]. For each word, one can count the number of segments, with at least one occurrence, in the base set (denoted $df_B$ for document frequency in the base set) and similarly in the counter sample ($df_C$). Only the presence or absence of a term is important not its repetition inside a segment. To obtain the proportion of segments in the base set with this term, one can divide the $df_B$ value by the number of segments (denoted $n_B$). For the counter set, we count the proportion of segments without this term as $(n_C - df_C)/n_C$.

A good discriminating term occurs in many segments in the base set and rarely in segments extracted from the counter set. Therefore, the resulting Zeta score, denoted Z(t) for the term $t$, is computed according to Eq. 7.1.

$$Z(t) = \frac{df_B}{n_B} + \frac{n_C - df_C}{n_C} \tag{7.1}$$

When a term appears in all base segments ($df_B = n_B$), its proportion is 1 for this sample. If it never occurs in the counter set ($df_C = 0$), the resulting $Z(t) = 1 + 1 = 2$. This value corresponds to the maximum and this term perfectly discriminates segments belonging to the base set over the rest. When a term occurs in all segments, then $df_B = n_B$ and $df_C = n_C$. The resulting $Z(t)$ value is $1 + 0 = 1$. This value is associated with the most frequent word-types, appearing in all textual chunks such as *the*, *of*, *in*, *is*, etc. Finally, when a term occurs only in segments belonging to the counter set ($df_C = n_C$), the $Z(t)$ value $= 0 + 0 = 0$.

[1] Even if the term *test* is used to denominate this strategy, it is not a real statistical test.

The Zeta score can vary from 0 for words appearing only in the counter set to 2 for terms occurring only in the base set. One can see the value 1 corresponds to a border between the two alternatives.

Based on the Zeta score, the entire vocabulary can be sorted from the largest to the smallest value. The terms depicting the highest Zeta values are reported in Table 7.1 for the three authors of the *Federalist Papers*. In this computation (repeated three times, once for each author), each segment contains 2000 tokens as suggested by [11] and the counter set contains the other two authors. Clearly, some words having a middle frequency can be associated with one writer such as *upon* for Hamilton or *fully* for Madison.

**Table 7.1** Five most discriminative terms according to the Zeta value for the three authors

| | Hamilton | | Madison | | Jay | |
|---|---|---|---|---|---|---|
| Rank | Zeta score | Term | Zeta score | Term | Zeta score | Term |
| 1 | 1.762 | upon | 1.444 | existing | 1.8 | would |
| 2 | 1.597 | community | 1.397 | fully | 1.8 | in |
| 3 | 1.452 | matter | 1.385 | among | 1.8 | they |
| 4 | 1.449 | kind | 1.382 | according | 1.8 | it |
| 5 | 1.419 | intended | 1.382 | clearly | 1.8 | been |

Usually, and as indicated previously, the words having a large Zeta value correspond to the studied stylistic markers. Antonia et al. [11] suggest to compute the Zeta value for word $n$-grams instead of isolated words, but only for short $n$-grams such as bigrams or trigrams. Longer sequences occur less frequently and tend to generate more erratic counts. Of course, the selected features can also be letter $n$-grams or other ones such as personal pronouns, wordlists of emotional expressions, emojis, etc.

The next step is to define the set of stylistic markers that can be associated with the base set. To reach this, the top $m$ terms having the highest Zeta scores can be chosen. Similarly, the top $m'$ terms with the lowest Zeta scores form the stylistic indicators for the counter set. No precise rules have been formulated to specify the value $m$ or $m'$ but usually $m = m'$. Of course, one must verify that the top $m$ terms must present a Zeta score larger than one. Similarly, the $m'$ words must have a Zeta score smaller than one. In practice, depending on the Zeta score distribution and the vocabulary size, one can consider values between $m = 200$ and 2000. For example, Antonia et al. [11] propose fixing $m = m' = 500$.

Based on these two wordlists, one can count for each segment the proportion of terms appearing in the set having the highest Zeta values ($x$-axis) or in the set containing words with low Zeta values ($y$-axis). These two percentages specify the position of this segment in the related scatterplot. The attached label is provided by the origin of this chunk (base or counter sample).

For example, in Fig. 7.1, the 22 chunks extracted from the base set (Madison's articles) are depicted with red diamonds and the counter set by 67 blue rounds (texts

written by Hamilton and Jay). The resulting scatterplot clearly shows two distinct clusters of points based on segments having 2000 tokens and considering $m = m' = 80$ terms.

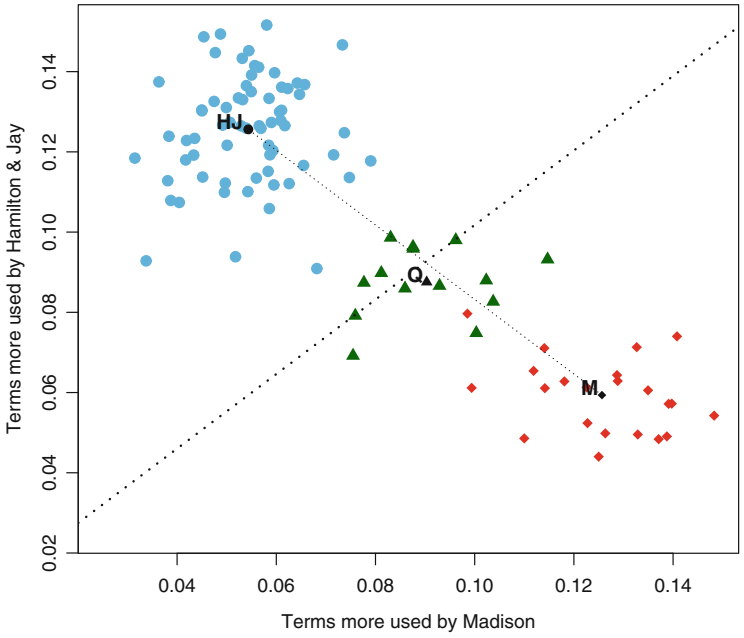


**Fig. 7.1** Zeta representation of the papers written by Hamilton and Jay and Madison (segment size = 2000, wordlist length = 80)

From each cluster, the centroid or gravity center is computed and displayed as a typical member of the group. In Fig. 7.1, these centroids are displayed with the letters "M" or "HJ." As shown in the figure, a light dotted line can be drawn to join these two centroids. To indicate the border between the two groups, a bisector line can be drawn perpendicular to the line joining the two centers. In Fig. 7.1, this limit is represented by a strong dotted line.

Then one can add the segments belonging to the unseen disputed texts performing the same operation. The proportion of terms appearing in the stylistic set of the base author ($x$-axis) and in the counter set ($y$-axis) determine its coordinates. The disputed chunks are represented by 13 green triangles in Fig. 7.1. In addition, the centroid of the disputed segments is reported with the letter "Q." Evidently in Fig. 7.1, all green triangles occur between the two clusters, but slightly closer to Madison's group.

With the border shown with a dotted line, the reader could have the impression that this limit could be applied to classify texts according to their author. This identification principle was applied with the SVM model (see Sect. 6.3) that defines

a linear border between the two possible regions. As mentioned by Rizvi [319], within the Zeta scatterplot not all regions are fully reliable. As soon as a point appears relatively distant from all undisputed points, it is rather difficult to assign it to one of the two categories with any certainty.

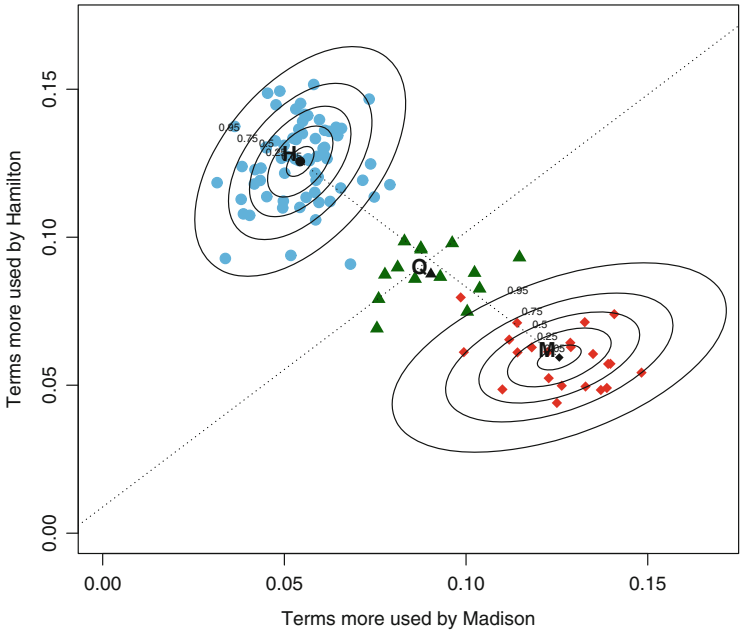


**Fig. 7.2** Zeta representation with two bivariate Gaussian distribution

To illustrate more clearly this limitation, Fig. 7.2 represents the two clusters with the assumption that they are generated by two bivariate Gaussian distributions, the first one for Hamilton's and the second for Madison's chunks. The contour of both distributions indicates more clearly the region where an attribution could be made with some certainty. As one can see, the majority of the disputed segments do not clearly belong to one of the delimited regions of the two possible authors (under the assumption that the bivariate Gaussian closely reflects the real distribution). Another real application of the Zeta test can be found in Sect. 8.5.

## 7.2 Compression Methods

To measure the intertextual distance between two texts, various compression-based approaches have been suggested. Using such methods, different text categorization problems can be solved [253, 393] as well as authorship attribution questions [24,

58, 147, 148, 288, 354]. The basic concept is to apply a compression method to measure the intertextual distance (or similarity). After defining such a distance measurement, the closest category profile (or instance) with the query text determines the proposed attribution. In its simplest form, this strategy corresponds to the 1-NN method with an intertextual distance derived from a compression algorithm.

To define such a distance, the underlying idea is the following. Assuming that Text A was written by the first author and Text B by a second unknown one, the notation $C(A)$ indicates the compressed size of Text A measured in bytes and similarly with $C(B)$. After concatenating the two texts, $C(AB)$ indicates the size of this union. Based on these values, the intertextual distance between A and B, denoted $NCD(A, B)$, is given by Eq. 7.2 [58, 241].

$$NCD(A, B) = \frac{C(AB) - Min\,(C(A), C(B))}{Max\,(C(A), C(B))} \tag{7.2}$$

This formulation indicates the normalized compression distance (NCD). This value is always positive in the range 0 to $1 + \epsilon$ ($\epsilon$ is a small positive value (e.g., 0.1) due to the imperfection of the compression algorithm). When the two texts have been written by the same author, the distance NCD should be small. For example, assuming that $C(B) > C(A)$, Eq. 7.2 becomes

$$NCD(A, B) = \frac{C(AB) - C(A)}{C(B)} \tag{7.3}$$

and the distance $NCD(A, B)$ could be interpreted as the improvement provided by the compression of Text A (knowing its frequent terms and expressions) for compressing B. When the contents of A and B are similar, the distance is small because Text A provides pertinent information about the used vocabulary and their term frequencies. Therefore, one can infer that both texts have been authored by the same person.

As an alternative, one can choose the conditional complexity of compression (CCC) defined by Eq. 7.4. In this computation, one measures the additional number of bytes required to represent Text B, knowing the size of Text A. As for the NCD values, a small number is an evidence in favor of a single author for both texts. However, as for the previous function, it is not clear what *small* really means and a comparative basis is mandatory.

$$CCC(A, B) = C(AB) - C(A) \tag{7.4}$$

As a third variant, one can apply the compression-based cosine (CBC) reported in Eq. 7.5. In this case, the distance is computed as 1—the similarity score obtained by a cosine-based formulation.

$$CBC(A, B) = 1 - \frac{C(A) + C(B) - C(AB)}{\sqrt{C(A) \cdot C(B)}} \tag{7.5}$$

As for the nearest-neighbor method (see Sect. 6.1), the identification of the true author of a disputed text can be determined using an instance-based or a profile-based approach. In the first case, the similarity is computed between the disputed document and each text in the training sample. The label of the closest text establishes the proposed author. In another way, one can consider the $k$ closest texts and the most frequent label defines the returned category. With the profile-based approach, all texts belonging to a given class are concatenated to form the general profile of that class. Then the closest profile determines the assignment to be given to the disputed text.

In a more recent study, Halvani et al. [148] propose using a compression-based method for solving the verification problem. To verify whether or not the same author wrote the query Text Q as well as a sample of texts denoted $A_1, A_1, \ldots, A_n$, the first step is to identify the text $A_{min}$ having the smallest similarity according to Eq. 7.6.

$$A_{min} = \text{Arg}\min_{A_j} \left(CBC(Q, A_j)\right) \tag{7.6}$$

$$S_{min} = \min_{A_j} \left(CBC(Q, A_j)\right) \tag{7.7}$$

The similarity value performed with the closest text $A_{min}$ is represented by the variable $S_{min}$ (see Eq. 7.7). A small $S_{min}$ is an indication that all texts have been written by a single author. However, a large $S_{min}$ value would contradict this conclusion.

In a second step, one needs to verify whether or not this $S_{min}$ value is small enough to confirm a unique author. To achieve this, the mean similarity is computed between all texts $A_j$ and $A_{min}$. This average is denoted $S_{avg}$ as shown in Eq. 7.8.

$$S_{avg} = \text{Mean}_{A_j} \left(CBC(A_{min}, A_j)\right) \tag{7.8}$$

As a decision rule, if $S_{min} < S_{avg}$, then one can assume that the query Text Q was written by the same author as the sample $A_j$. Otherwise, the answer is negative; Text Q is authored by another person than the author of text $A_j$.

In this distance evaluation, one can assume that all texts (Q and $A_j$) have a similar length. This constraint is not a strict one but tends to produce better effectiveness. For example, Halvani et al. [148] indicate a difference in size in the ratio 1:3 without significantly modifying the overall results.

As compression algorithm, it appears that GZip tends to perform better than BZip or PPM when considering an instance-based approach [288]. However, with a profile-based method, the BZip algorithm seems to offer better performance.

With this family of authorship attribution models, the preprocessing could be rather limited (e.g., to simply remove tags). One does not need to specify the best feature subset or even to select the features. In addition, the proposed method works for all languages without any specific adaptation (e.g., tokenization). In some

cases, the overall performance could be higher than some of the state-of-the-art approaches [278].

These attribution methods are not without drawbacks. First, they consider the entire vocabulary with their frequencies and even words appearing once or twice. Moreover, those words have a larger contribution in determining the size of the compressed text. According to Zipf's law (see Sect. 2.3), they correspond to around 50% of all word-types. This aspect contradicts the linguistics theory emphasis on frequent and ubiquitous items as useful stylistic markers for identifying an authorial voice. Moreover, the similarity measure between two texts can be computed according to different formulations and the most effective compression algorithm is not clearly defined. Finally, it seems that applying an instance-based approach tends to perform better than a profile-based one. Depending on the choices for these three parameters, the evaluation performance could be significantly different [288].

## 7.3 Latent Dirichlet Allocation (LDA)

To solve the authorship attribution problem, one can assume that the subjects or topics of the disputed text could be useful in identifying the correct author. This idea is not fully new because when considering more than let us say 100 MFWs, numerous words are no longer functional ones but indicate the underlying topics. For example, and as shown in Appendix A.2, in the top 50 MFWs from the *Federalist Papers*, one can find the content word *states*, *government*, *state*, or *power*. With the LDA model, the novel aspect is the fact that the topics are explicitly taken into account.

Latent Dirichlet allocation (LDA) [35, 36] is an extended version of the probabilistic generative model suggested by Hofmann [159]. In this framework, the objective is to model the creation of a set of documents. Each text is viewed as composed of a mixture of topics. Of course, a given document may cover only a single topic, but this is more the exception than the norm. For example, the first one might cover intensively Topic 1, present some aspects related to Topic 2, and marginally to Topic 3. An equal mixture of Topics 1 and 3 might engender the second article, etc. Therefore, each document can be modeled as a distribution over the $k$ topics, where $k$ defines the maximum number of topics. In LDA, this $k$ value is fixed and defined a priori. As a generalization, each document is viewed as generated according to all $k$ topics with different intensities (e.g., when a given topic does not appear in a document, its intensity is set to 0).

In this probabilistic generative view, the term *topic*[2] does not correspond to a symbolic subject heading such as "Quantum computing" or "Medieval poem" but

[2]In LDA the meaning of the word *topic* does not correspond directly to subject or theme but to a wordlist with their occurrence probabilities. Thus we prefer to write it in italics to underline this specific interpretation.

defines a specific word distribution. In other words, each *topic* is a list of word-types with their occurrence probabilities (see examples in Table 7.2). When inspecting the vocabulary of a given corpus, one can find some specific word-types strongly related to a single subject. Therefore, such terms will only appear in that particular distribution (or *topic*). Usually, however, a word tends to cover different semantic fields such as the word Java (a coffee, an island, a programming language, a dance, etc.). In such cases, the word may appear in different *topics* to model its polysemy. Finally, when a word is a function word or simply when it occurs in almost all texts (e.g., *the*, *of*, *in*, *was*, *would*, etc.), each *topic* will include it usually with a different occurrence probability.

**Table 7.2** Five most frequent terms according to five *topics*

| | Topic 1 | | Topic 3 | | Topic 4 | | Topic 11 | | Topic 13 | |
|---|---|---|---|---|---|---|---|---|---|---|
| Rank | Term | % | Term | % | Term | % | Term | % | Term | % |
| 1 | the | 7.95% | the | 8.01% | she | 4.72% | he | 8.73% | the | 8.53% |
| 2 | of | 2.76% | in | 4.27% | to | 4.21% | his | 6.45% | said | 3.45% |
| 3 | to | 2.57% | to | 3.74% | her | 4.20% | to | 4.29% | was | 3.44% |
| 4 | a | 2.56% | of | 3.31% | a | 3.96% | the | 3.89% | to | 3.42% |
| 5 | Scotland | 2.44% | and | 2.58% | and | 3.45% | a | 3.66% | of | 2.76% |

As an example extracted from a newspaper collection, Table 7.2 reports the top five MFWs from five *topics* [337]. The determiner *the* is the most frequent word for Topic #1, #3, and #13 but not with exactly the same occurrence probability. In Topic #11, *the* does not appear with a very high probability compared to the other *topics*, while in Topic #4, it does not appear among the top five. In addition, this *topic* might certainly more related to a feminine subject (frequent used of the pronoun *she* and *her*).

Given as input the number $k$ of *topics* and a set of documents, LDA will determine the most likely *topic* model corresponding to this data. In this formulation, the hidden structure is the *topics* (defined by their word distributions) and the distribution of the *topics* in each document. Thus, LDA needs to first estimate the probability distributions over words associated with each *topic* and then the distribution of *topics* over documents. Of course, the objective is to discover the hidden structure that best explains the observed words and documents [36]. In this optimization procedure, the word position in the document or inside a sentence is not taken into account (bag-of-word assumption), as well as the document order in the corpus.

Of course, some preprocessing of the corpus can be applied, for example, to ignore word-types having a frequency smaller than a threshold (e.g., 5) or appearing only in a few documents. In other applications, the vocabulary could be defined, for example, by considering only the 500 MFWs (words, $n$-grams of words or of letters) or limited to terms appearing in a subject heading catalogue.

How can we apply this model for solving an authorship attribution problem? As a first implementation, one can consider that one *topic* (i.e., a given word distribution) corresponds to one author (see examples in Table 7.2). After estimating the underlying distributions, one can infer the mapping between the *topics* and the author names by considering the topics distribution over the text sample (for which the author of each document is known). Using these distribution estimates, the system can infer the *topic* distributions of a new and unseen document. For example, if the doubtful text corresponds to 50% of Topic 3, 30% of Topic 1, and 20% of Topic 4, the proposed author is the one related to Topic 3.

In a second implementation, the number of *topics* could be larger than the number of possible writers. It is more realistic to assume that an author could write about a few themes. Knowing the set of texts written by a given writer, the *topic* distribution over them could be averaged to create an author profile. Thus, for each possible writer, a *topic* distribution is provided. In our experiments with 20 authors and $k = 40$ to 80 topics [337], we usually found that an author could be characterized by a few principal *topics* (between 1 and 4), the remaining ones appearing with low or even very low probability. In a related study, Rosen-Zvi et al. [323, 324] propose an extension to LDA to directly take account of the distribution of the authors over the topics.

As soon as the *topic* distribution is known for a disputed document, one can measure the distance between two *topics* distributions, i.e., the one reflecting the author, the second extracted for the disputed document. To achieve this, the Kullback–Leibler divergence (see Sect. 3.3) could be applied and the shortest distance indicates the most probable author. In a preliminary study using LDA in authorship attribution [337], the effectiveness was relatively high (between 80 and 90%) when faced with newspaper articles (median length around 700 words) written by 20 possible columnists.

Of course, other applications could be achieved using LDA as a way to represent the stylistic or, more generally, the vocabulary change over the time. As an example, one can cite [130] who illustrate the temporal evolution of the American sociology by examining the titles and abstracts of the journal *American Journal of Sociology*. Another example is provided by [328] about the *State of the Union* addresses (see also Chap. 10).

## 7.4 Verification Problem

In the different authorship attribution issues described in Sect. 1.3, the verification question seems the simplest one. Its formulation is the following. Having a query Text Q and an Author A, the question is to know whether or not A wrote this text. The expected answer is binary (yes or no) and some justification could complement this proposed decision [214, 215, 218]. Of course, to provide a response, the machine should analyze a sample of texts written by the proposed Author A to determine his authorial style.

Why is this question a hard problem? One can simply compute an intertextual similarity between the disputed Text Q and the different texts included in the sample written by A. If the similarity value (or the distance measure) is high enough, one can conclude that Document Q was written by A; otherwise, the answer is negative [1]. However, as discussed in Sect. 1.2, the style can be explained by different factors, and only one of them is the author. Thus a pertinent sample should correspond to texts written in the same genre as Text A, with similar topics and published during the same period. As a concrete example, one can consider this question with Shakespeare. Suppose we want to verify whether or not F. Bacon is behind some plays attributed to Shakespeare. It will be impossible to have plays written by Bacon to verify our assumption in optimal conditions. What can be collected are texts written by Bacon but corresponding to distinct text genres.

In fact, the text genre generates a noteworthy style variation. Writing a crime novel or a play implies distinct styles, and the second could be written in prose or verse, increasing the dissimilarity. In addition to this source, one can consider variations in topics, target audience, or communication medium. Even when limited to texts written by the same author, a large time gap between the publication date of two texts can explain stylistic divergence. Finally, a style is not fully stable throughout the author's life [121]. Therefore, all these factors can decrease the similarity measure even when the two texts have been written by the same author [50, 171].

The solutions proposed to track this verification question show three new aspects, namely the concepts of *impostors*, the *second-order similarity* measure, and the *unmasking* strategy. Starting with the impostors' method.

**Impostors** To solve the verification problem, one can determine a set of potential authors, called impostors, who could also include the possible true author of the disputed text. This selection is not so easy because the impostors' texts must have been written in the same (or similar) text genre as the query text, during the same period, possibly on similar topics and with the same target audience. Moreover, one can never be 100% sure that all possible impostors have been selected. Consider the following example. One must identify the author of a set of threatening e-mails. In this case, it is hard to define an exhaustive list of all potential writers and to obtain a sample of texts for each of them. But all applications do not present these difficulties and in some cases the set of impostors could be easier to define.

When generating the impostors, one should consider the style similarity with the main suspect, and the higher this similarity, the better the quality of the impostor. This could be achieved when considering impostors writing in the same text genre as the sample of texts authored by A. Of course, high similarity with other factors such as the time period or the target audience will increase the impostors' quality. The second question is to determine the number of impostors. One can assume that a higher number implies a better solution. Usually, however, the most important factor is to choose impostors presenting a high stylistic similarity with the suggested author [215].

**Table 7.3** Impostors' algorithm for solving the verification problem

```
Given a query Text Q, a feature set FS, and
    some author profiles C
C = C ∪ A
Repeat k times { # e.g., k = 100
 Randomly select 50% of the features in FS
 Compute sim(Q, Ci) for all known author profiles in the set C
 Increase the score of the author appearing in the first position
    in the ranking list
}
If the highest score over all candidates in set C is A,
    and if score(A) > δ
 return "A is the true author"
 else return "Unknown author, but not A"
```

Assume that a set of impostors (denoted $C$) has been determined with a sample of their texts for each of them. From the samples written by A and not by A, a binary learning model (SVM, logistic regression, naïve Bayes, Delta, etc.) can be trained to discriminate texts authored by A vs. not-A. Of course, the impostors' group $C$ represents the not-A profile. In a final step, the query text Q is classified as A or not-A. This last response is usually formulated as *unknown author, but not A*.

As an effective variant, instead of considering Text Q as a single entity, one can split it into $k$ non-overlapping chunks or segments (e.g., from 500 to 2000 tokens). Instead of having a single assignment, the binary classifier is applied $k$ times. When the majority of these $k$ chunks is attributed to A, the final decision is A, otherwise not-A. When choosing this alternative, the size of the query text must be large enough to generate many $k$ segments.

The verification problem could also be solved in another way [220] as reported in Table 7.3. With the chosen classifier, not all features are used to determine whether the author is A or not-A, but a random sample of them. During an iteration, a percentage of the features are randomly selected from the full attribute set (e.g., 50% [215]). Based on this reduced feature set, the classifier determines whether the author of the query Text Q is A or not-A. After $k$ iterations, if the proportion of assignments to A is larger than a predefined threshold (denoted $\delta$ in Table 7.3), the final decision is A, otherwise not-A. Of course, to apply this strategy, one needs a large feature set (e.g., varying length character $n$-grams).

Instead of a binary classifier, the system could take account of $r$ candidates, author A, and the $r - 1$ impostors. In such cases, the classifier could rank the $r$ possible authors at each iteration. The final decision (A or not-A) depends on the $k$ ranking lists [215] as described in the algorithm shown in Table 7.3.

**Unmasking** As another interesting strategy to discover whether or not A is the true author of the query Text Q, one can apply the unmasking method [218]. To describe its simplest form, let us assume we have a sample of texts written by A and

similarly for a set of impostors. To simplify the presentation, let us assume that only two impostors are considered, denoted B and C. For each author, a classifier (e.g., SVM) is built to discriminate between a given author and not this author (e.g., B vs. not-B). Using an evaluation methodology (e.g., 10-fold cross-validation), and for each possible author (namely A, B, and C), we evaluate whether or not this author is the true writer of Text Q. This first evaluation is represented by the first point in Fig. 7.3. However, instead of performing this evaluation once, it will be iterated $k$ times. For example, in Fig. 7.3, this evaluation was performed eight times. At each iteration, $r$ features are removed (e.g., $r = 3$), not randomly, but the most pertinent features able to discriminate between the proposed author (A, B, or C in our example) and those related to text Q. One can also remove the $r$ most weighted negative features.

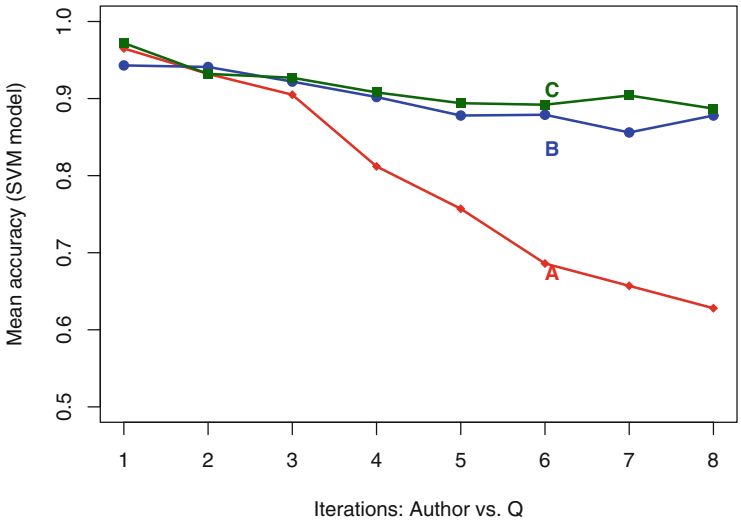


**Fig. 7.3** Evolution of the mean performance with three possible authors

One possible outcome of this evaluation strategy is depicted in Fig. 7.3. Over all iterations, the mean accuracy rate could present a relatively stable line when considering the profile of Author B or C. For Author A, a clear degradation is detected. After six iterations, a clear difference in the mean effectiveness can be observed. This disparity can be explained by the removal of many features strongly related to the authorial style of A. When the query text is written by the proposed author (A in our example), iteratively removing useful features strongly related to A and Q means that the identification of the true author (A) is more and more difficult. The performance curve is therefore decreasing for the true author, not for the other ones.

**Second-Order Similarity** The different distance or similarity measures discussed in Sect. 6.1 are called first-order functions. The second-order similarity measure was suggested by Koppel and Seidman [214] as a method to solve the verification problem. Let us assume that we have a set of texts denoted $C = \{c_1, c_2, \ldots, c_m\}$ written by different impostors and a second set $A = \{a_1, a_2, \ldots, a_n\}$ of texts that could have been written by a proposed author A'. We are not sure that all texts in set A are really written by A'. From sets A and C, one can extract a large feature set (e.g., varying length character $n$-grams).

The basic idea (see Table 7.4) is to compute the similarity between two text representations extracted from set A and denoted $a_i$ and $a_j$. One can select a similarity measure (see Sect. 6.1) and compute the similarity between them and denote this similarity value as $sim$ (in short for $sim(a_i, a_j)$, a first-order similarity). When the same author is behind these two texts, this value must be high.

**Table 7.4** Second-order similarity algorithm to solve the verification problem

For a set of text $A = \{a_1, \ldots, a_i, \ldots, a_j, \ldots, a_n\}$,
  a feature set FS, and a set of texts $C = \{c_1, c_2, \ldots, c_m\}$
  For each pair $(a_i, a_j)$ {
   $sim2(a_i, a_j) = 0$
   Repeat r times {
    Randomly select 50% of the features in FS
    Compute $sim(a_i, a_j)$ # a first-order similarity
    For all texts $c_k$ for $k = 1, 2, \ldots, m$ do:
     compute $sim(a_i, c_k)$ and $sim(a_j, c_k)$
    If $(sim(a_i, a_j) \times sim(a_i, a_j) > \max(sim(a_i, c_k)) \times \max(sim(a_j, c_k)))$ )
     $sim2(a_i, a_j) = sim(a_i, a_j) + 1/r$
   }
  } # Take the next pair of texts
 $dec(a_i)$ = Aggregate $(sim2(a_i, a_j))$ # e.g., the median
 if $(dec(a_i) > \delta)$
  return "$a_i$ was written by the proposed author"
  else return "$a_i$ was written by an unknown author"

But what does a high similarity value mean? To verify this, the similarity is also computed between, respectively, $a_i$ and $a_j$ and all $m$ texts belonging to set C. Only the largest similarity with items in set C for both $a_i$ and $a_j$ is retained. We multiply them. If this product is smaller than $sim$ to the power 2, one can conclude that $sim$ presents a high value and this is evidence that $a_i$ and $a_j$ have been written by the same person. As shown in Table 7.4, this procedure is then repeated $r$ times (e.g., $r = 250$).

After considering all pairs of documents $(a_i, a_j)$, one can conclude whether or not each text $a_i$ was really written by A'. To make this final decision, one needs to aggregate all the second-order similarity values ($sim2(a_i, a_j)$ in Table 7.4)

computed for $a_i$. As we do not have any guarantee that all texts in A have been authored by A', using the arithmetic mean as an aggregate operator is not a very good choice. It is known that the presence of an outlier could notably affect the mean. As a more robust approach, one can suggest using the median. As a variant, one can compute the mean but not over all texts in A, but only for the $k$ closest ones [214]. Finally, when the aggregated value is larger than a specified threshold (denoted $\delta$), the system answers that Text $a_i$ was authored by the proposed writer; otherwise, this text was written by an unknown person.

As another simple unsupervised approach to solve the verification problem, Kocher and Savoy [208] suggest computing the distance[3] between the query Text Q and the profile of the proposed author A' (with his sample of writings). The essential question is to determine if this resulting value is high and then one can conclude that A' was not the true author of Q or the reverse. As for other stylometric problems, a comparative basis is required. In this case, it is achieved by considering $r$ times up to $m$ impostors with texts having a similar genre as the query Text Q (See algorithm described in Table 7.5).

**Table 7.5** Kocher's algorithm to solve the verification problem

Given a sample of texts A written by A',
  a query text Q, a set of impostors C (with texts)
Use Q and A to generate the feature set FS
Generate the profile A' from A and the profile from Q
$\Delta_0 = dist(A, Q)$ # Using the Manhattan distance
Repeat r times {
 Generate the profile for the $m$ impostors $C_i, i = 1, 2, \ldots, m$
 $\Delta_k = Min_{i=1,2,\ldots,m}\{dist(Q, C_i)\}$
}
$\Delta_m = mean(\Delta_k)$
if $(\Delta_0/\Delta_m < 0.975)$ return "A' is the author of Q"
if $(\Delta_0/\Delta_m > 1.025)$ return "A' is not the author Q"
 else return "don't know"

In summary, the verification issue is still a hard problem and research papers indicate accuracy rates varying between 70 and 85% depending on the test collections and parameter values. We need to keep in mind that the usual condition occurring in a forensic case is not the ideal context, namely having long texts (10,000 or at least 5000 words). Moreover, for an effective solution, one needs several texts from the proposed author and from all possible impostors (or at least from many of them). In addition, it is important that the available texts are written on the same text genre, and hopefully with similar topics. Of course, the number of impostors could also play a role in the overall effectiveness of the proposed solution.

[3] Using the distance instead of the similarity, we need to consider small values as evidence that the two texts are written by the same author.

## 7.5 Collaborative Authorship

Usually, one can assume that each document has been written by a single author. When considering blog posts, poems, novels, or newspaper articles, this underlying assumption is valid most of the time. In other text genres, the presence of multi-authors is the norm not the exception, for example, for scientific papers. In such cases, however, the writing itself could be done by a single author, while others could be in charge of collecting the data, elaborating the protocol, implementing the algorithm, performing the statistical analysis, etc. Of course, a scientific article could have also been written by several authors, usually each of them has been in charge of one or more sections. Thus, the definition of a *collaborative authorship* means that two or more authors have *written* the final text. Therefore, other possible forms of collaboration are ignored such as jointly elaborating the scenario or the main figures of a novel. In literature, one can find such writing collaboration, for example, when the main author died without being able to finish a novel or to continue a series of books. For example, L. Frank Baum created the main figures of the Oz world and wrote the first fourteen books in this series. The question that remains, who wrote the 15th (*The Royal Book of Oz*) [33]? In this section, however, the focus is on a single document that could have been written by $k$ authors (for $k = 1, 2, \ldots$).

As this collaborative question reflects different situations, various solutions have been suggested, for example, in the case where the number and names of the collaborative authors are known. In this case, text samples of each possible author are available and the rolling Delta can be applied. When only the number of authors is known without any text samples from them, some unsupervised ad hoc approaches could be suggested. Finally, when even the number of writers is unknown, the problem is harder and is investigated through some CLEF PAN evaluation campaigns [429]. Let us start with the first context.

**Rolling Delta** An interesting extension of the Delta method (Sect. 3.1) called rolling Delta was proposed by [99, 333]. The main idea is not to represent the entire novel by a single vector but subsets of that work, each with its own vector. To subdivide a document, one can take account of its logical structure (e.g., each chapter forms a separate entity) or according to a window (of fixed size) generating chunks of a text containing $k$ words or sentences (e.g., $k = 5000$ words). With this second solution, the segments can be generated in a non-overlapping way (the intersection between them is nil). Thus, the first chunk starts from the first word to the word appearing at position 5000, while the second segment begins at position 5001 to 10,000, etc. With an overlapping generation, the length of the overlap must be specified (e.g., a step size of $d = 100$). In this case, the first chunk starts with the 1st word to the 5000th, and the second chunk begins at position 101–5100, etc. But why decompound a text like this?

With this decomposing process, the text is then viewed as a stream of chunks (the rhetoric timeline) from which one can detect stylistic change. Each segment in turn forms the query text. Then the selected classifier (e.g., the Delta method, SVM,

logistic regression) attributes each chunk to one of the possible authors (closed-set assumption). This assignment could be associated with a degree of certainty (e.g., the distance to the class boundary with the SVM model).

As an example, Eder [99] has applied this strategy to the French allegorical poem called *Roman de la Rose*. This work was written by two authors, the first part by Guillaume de Lorris (around 1230) and then completed by Jean de Meun (1275). According to scholars, the first part contains 4058 lines (around 50,000 words), while the second regroups 17,624 lines (for around 218,000 words).

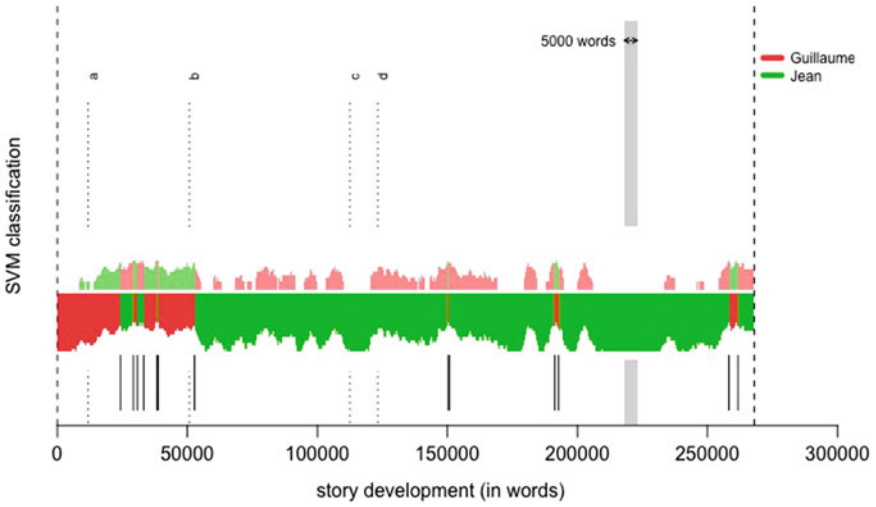


**Fig. 7.4** Rolling Delta applied to *Roman de la Rose* with SVM and 100 MFWs (©2016 Maciej Eder, GPL-3.0)

Using an SVM classifier and 100 MFWs as stylistic markers, the rhetoric timeline of this poem is depicted in Fig. 7.4 [99].[4] To achieve this, the work was subdivided into overlapping windows of 5000 words with $d = 100$. To determine the authorial voice of the two writers, the training corpus contains two parts, namely from the beginning to the letter "a" and between the letters "c" and "d" (see Figure 6). The attribution of each segment to both authors is provided by the bottom ticks, the longer the tick, the higher the certainty that the proposed attribution is correct. The top ticks show the degree of certainty associated with the second writer.

As one can observe, the style change denoted by the letter "b" in Fig. 7.4 was correctly identified. Of course, not all segments are perfectly assigned to their respective author, but the large majority of them, especially those written by Jean de Meun, are properly attributed.

[4]This figure was produced with the `stylo` package and the associated datasets, freely available on the Internet, see `www.computationalstylistics.github.io`.

As another example, this methodology can reveal the precise contribution of each author when faced with a collaborative work, such as with Conrad (1857–1924) and Ford (1873–1939) with the novels *The Inheritors* (1901), *Romance* (1903), or *The Nature of a Crime* (1909, 1924) [99, 333]. Studying a collaborative work in the Middle Ages raises more challenges than working with Modern English novels. Obviously, detecting the authorship before the contemporary period remains a more complex question [204] due to the spelling and grammar variability and the uncertainty associated with the true author.

**Ad hoc Approaches** When applying the rolling Delta, one needs to know the number of authors indicated by $k$ as well as samples of texts for each of these $k$ writers. These text samples are used to generate distinct author profiles. With the *Federalist Papers*, we precisely encounter this situation. Articles #18, #19, and #20 have been jointly written by Hamilton and Madison but with no information to precisely specify the exact contribution of either. In other cases, we simply do not have a text sample for each possible author. Thus, the rolling Delta cannot be applied.

In this more complex situation, Akiva and Koppel [7] propose an interesting method to solve this question under the unique assumption that the number $k$ of authors is known. Not having texts written by the possible authors means that a *supervised* approach is impossible. The required information to learn each individual style is simply not available. The noteworthy idea is to transform an *unsupervised* problem into a supervised one.

The general algorithm is described in Table 7.6 [7]. As input, a (long) text written by the different authors is provided as well as the number $k$ of authors. For example, this document could be a long blog written by two authors, a set of scientific papers or the concatenation of numerous newspaper articles. To define the split points between two authors, one can assume that they never appear inside a sentence, or in other words, each sentence is written by a single author.

The first step of the proposed approach is to chunk the text into segments having the same number of sentences (e.g., 40). Next, each chunk is represented by a binary vector composed of the $m$ MFWs. In this surrogate, the presence and absence of the selected terms are taken into account, not their frequencies. This kind of representation was used in the multivariate Bernoulli model (see Sect. 6.2). This choice could be justified by the fact that the main objective is to discriminate between authors. In this view, the binary approach emphasizes the words used or ignored by the different authors, not on their repetitions. Of course, each segment could contain sentences written by a single author or by more than one. At this stage, this distinction is not possible.

In Steps #3 through #5, the similarity between all segments is computed to produce the information needed to generate the $k$ clusters. After this, we hope that each cluster mainly contains segments written by a unique author. Therefore, we could admit that each segment belonging to a cluster was written by that author, with some additional noise (sentences scripted by a second writer). In other words,

**Table 7.6** Identifying passages written by each author

```
Given a text, and k the exact number of authors.
1. Chunk the text into segments of fixed length (e.g., 40 sentences).
2. Represent each segment by a binary vector (e.g., vector of 500 MFWs).
3. Compute the similarity between every pair of segments
    (e.g., Cosine).
4. Cluster the segments into k clusters (see [201]).
5. Each segment received the label by its cluster assignment
   to generate a labeled dataset.
6. Choose a classifier (e.g., SVM) to be trained with the segments
   and their labels.
7. Used the trained classifier to assign each sentence to one of the
   k clusters.
 If the assignment is achieved with a high confidence, lock it.
 Otherwise, assign the sentence according to previous and next
    sentence in the text.
```

we can now label the data and a supervised approach is then possible. Starting with unsupervised data, a sample of labeled segments has been generated.

In Step #6, $k$ author profiles are built from the segments included in each cluster. To achieve this, each text could be represented by a larger number of features considering not only the words but also other sources of evidence (e.g., $n$-grams of letters, POS, etc., see Sects. 5.1 and 5.2).

In Step #7, the trained classifier is then employed to classify each individual sentence belonging to the input text according to the $k$ possible authors. When the decision is taken with high confidence, one can grant this as a final decision. Otherwise, one can consider the short context of the input sentence $s$. If the achieved assignment of the previous and next sentence of $s$ belongs to the same class, this label is assigned to $s$. In other cases, the sentence $s$ could stay unassigned or its author could be identified according to a larger context (i.e., more than two sentences).

The overall effectiveness of the proposed solution depends on different factors such as the number $k$ of authors or the mean length of passages written by a single writer. Experiments performed by Akiva and Koppel [7] indicate a performance level close to 90% when faced with $k = 2$ authors and a mean number of sentences per author larger than 50. With $k = 3$, the overall effectiveness decreases to 80% or even 75% (with $k = 4$).

In conclusion, identifying the contribution of each author in a document written jointly by an unknown number of authors is a hard task. As shown previously, one can decompose this text into smaller chunks. For example, one can assume that each paragraph was written by a unique writer. Under this hypothesis, one can view each paragraph as a distinct text. In this case, the problem of the collaborative authorship is the same as the author clustering problem (see Sect. 1.6) for which some solutions

are possible (see [211]). Undergoing investigation inside the CLEF PAN evaluation campaigns [429] could provide more efficient solutions in the future.

## 7.6 Neural Network and Authorship Attribution

Recently under the deep learning paradigm, different neural network models have been proposed to solve difficult tasks such as image recognition [26]. In NLP, these kinds of approaches have been applied with success, for example, in speech or handwriting recognition, sentiment analysis of customers' comments, or automatic translation [132, 385]. But before describing some deep learning concepts in the next two sections, the basic notions of neural computing must be presented (a gentle introduction without mathematical background is provided by [202] and with examples in Python, consult [399]).

In this machine learning paradigm, the fundamental building block is an artificial neuron corresponding to a simple processing unit. As depicted in Fig. 7.5, a neuron is composed of an internal structure and has connections with other ones located either on the left or on the right. In Fig. 7.5, one can imagine five neurons on the left belonging to a previous layer and two in the next layer appearing on the right.

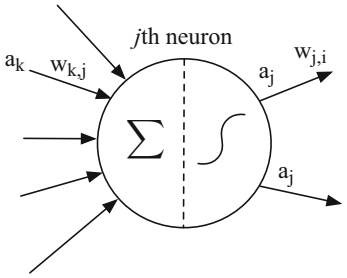


**Fig. 7.5** Representation of a single neuron

The work done by an artificial neuron is achieved in a sequence of three steps. First, the neuron computes a weighted sum of all its entries provided by the previous layer. This resulting value denoted $net_j$ for the $j$th neuron having $m$ connections with the previous layer is computed as follows:

$$net_j = \sum_{k=1}^{m} w_{k,j} \cdot a_k + b_j \tag{7.9}$$

in which $w_{k,j}$ is the weight associated with the connection between the neurons $k$ and $j$, and $a_k$ indicates the activation value transmitted by the $k$th neuron. When the weight between two neurons is positive, both neurons tend to be activated at the same time. With a negative weight, the first neuron inhibits the second. In Fig. 7.5, this function is symbolized by $\Sigma$ inside the neuron.

In addition to the activation values received by the previous layer, each neuron has a bias value, denoted $b_j$ for the $j$th neuron, to increase ($b_j > 0$) or decrease ($b_j < 0$) the computed weighted sum. This value can be viewed as a threshold to be reached to achieve a $net_j$ positive value.

In a second step, the neuron determines the activation value to be sent to the next level. This value is defined through an activation function presented in the right part inside the neuron in Fig. 7.5. As activation function, one can find the sigmoid (see Sect. 6.4, Fig. 6.12). In this case, the activation value denoted $a_j$ is defined between 0 and 1 depending on the value $net_j$. As a variant, the $tanh$ function can be applied. This function returns values between -1 and 1. As depicted in Fig. 7.6, when the $net_j$ is close to zero, the defined activation level is small and the effect of this neuron is rather limited. However, when $net_j$ is a positive value, the activation level is positive with a maximum reaching 1. Similarly, when $net_j$ is negative, the neuron will send a negative activation level to the neurons located in the next layer.

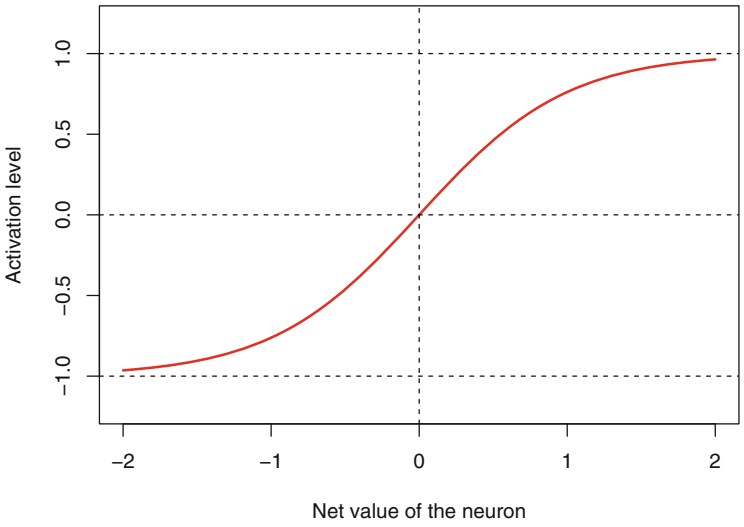


**Fig. 7.6** The tanh function

In a third step, the achieved activation value $a_j$ is propagated to the connected neurons belonging to the next layer. Then the same sequence of computations is performed by the next layer.

With such building blocks, a network of neurons can be organized into layers. In this perspective, a classical network called multi-layer architecture is composed of three layers called, respectively, the input, hidden, and output layers[5] as depicted

[5]Sometimes, the input layer does not count as a real layer, and in this case, Fig. 7.7 is composed of only two layers.

in Fig. 7.7. Such an architecture was proposed by [254] to solve an authorship attribution problem between Shakespeare and Fletcher. In another study, Tweedie et al. [410] adopted a similar neural network to identify either Madison or Hamilton as the true author of the 12 disputed articles of the *Federalist Papers*. In Matthews and Merriam's study [254], the input layer is composed of five neurons, one for each of the following functional words: *are*, *in*, *no*, *of*, and *the*. As input values, the authors proposed to compute the relative frequency of those five selected words. As a variant, the input layer could be composed of five terms (*did*, *no*, *no*, *to the*, *upon*) with the values corresponding to the ratio between the frequency of *did* divided by the frequencies of (did + do), and similarly with *no* and (no + not), *no* and (but + by + for + no + not + so + that + the + to + with), *to the* and (the), and *upon* and (on + upon).

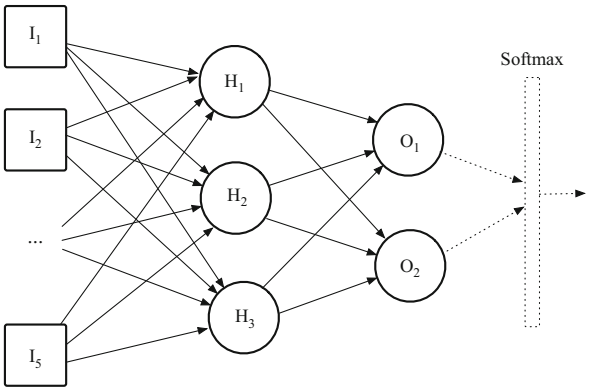


**Fig. 7.7** Representation of a neural networks with one input, one hidden, and one output layer

In Tweedie et al. [410], the input layer is composed of 11 function words, namely *an*, *any*, *can*, *do*, *every*, *from*, *his*, *may*, *on*, *there*, and *upon*. In this case, one can represent a particular example as a vector of 11 values corresponding to the occurrence rate per thousand of each selected word.

Usually, the input values correspond to the activation level for this layer. In this case, these values are directly spread through the network to the hidden layer. As a variant, the input values correspond to the *net* values that are then used to define the activation level of each input unit.

In the hidden layer, each neuron computes the net and their activation value determined by a sigmoid function as in [254, 410]. These values defined between 0 and 1 are sent to the output layer. In both studies [254, 410], the hidden layer is composed of three units as shown in Fig. 7.7. Finally, the neurons in the output layer are able to compute their net values and activation levels. In both applications, the number of neurons in the output layer is two because the number of possible authors is limited to two reflecting either Shakespeare and Fletcher [254] or Hamilton and Madison [410].

Using the sigmoid as the activation function, the output value for each neuron is in the range [0–1]. For example, the result could be 0.8 for the first neuron and 0.4 for the second and those two values do not reflect a probability distribution. When applying the $tanh$ as activation function, the achieved activation level is defined between $[-1–1]$ and the resulting values could be more problematic to interpret. Nowadays, an additional *softmax* computation is usually suggested in the final step. This is not mandatory and thus it appears with dotted lines in Fig. 7.7.

The *softmax* operator (for soft maximization) transforms a vector of values into a probabilistic distribution as defined by Eq. 7.10. In this formulation, $p_j$ indicates the probability associated with the $j$th neuron having an activation level denoted $a_j$. The denominator of Eq. 7.10 plays the role of a normalization factor.

$$p_j = \frac{e^{a_j}}{\sum_{k=1}^{n} e^{a_j}} \tag{7.10}$$

For example, representing the output layer by the vector $[0.7, -0.2, 0.3]$, the *softmax* function returns the probability distribution [0.481, 0.196, 0.323].

To achieve good predictions, the different weights $w_{k,j}$ and biases $b_j$ must be adjusted through a learning stage. Therefore, both a training sample and a learning algorithm must be provided. In both studies, one can identify articles or passages written without any doubt by one of the authors. In such cases, one can generate the input vector with the corresponding values and the correct output vector is defined as [1, 0] or [0, 1].

When the computed answer does not fit the correct one, the system must modify the weights associated with each link (e.g., $w_{k,j}$) as well as the bias values (e.g., $b_j$) of each neuron. Depending on the neural architecture, different learning algorithms can be applied. The classical strategy is called back-propagation [329]. The idea is as follows. During the prediction phase, the flow goes from the input layer to the output one. During the learning phase, the correction follows the reverse direction, from the output layer to the input. At the output layer, the error could be computed as the difference between the target value and the computed one. Based on the amplitude of this error, the modification of the weights can be performed. As other factors, the adjustment of the weight $w_{k,j}$ also takes account for the activation levels of the two neurons $a_k$ and $a_j$ and a learning rate (denoted by $\alpha$). This last component is a constant signaling how fast the change must be performed, and usually it is better to move by small steps (e.g., $\alpha = 0.001$). After modifying the weights between the output and the previous hidden layer, the process continues between the hidden and the previous ones up to reach the input layer.

After modifying the model for all instances in the training set (defining one epoch), the training stage is repeated for a given number of epochs or until a stability condition is reached (the weight changes are viewed as marginal).

Using this classification strategy, Tweedie et al. [410] were able to identify Madison as the true author of the 12 disputed articles. Analyzing four plays act-by-act, Matthews and Merriam [254] found that the *Two Noble Kinsmen* corresponds more to a Shakespearian style but with the clear collaboration of Fletcher (especially

in the second and third acts). For *Double Falsehood*, the Shakespearian influence is clearly present in the first act, while the rest of the play is weakly attributed to Fletcher. For the play *Henry VIII*, the results indicate a predominate Shakespearian stylistic voice but with less certainty in the last act. Finally, the work *The London Prodigal* corresponds more clearly to the Fletcher style.

## 7.7 Distributed Language Representation

More recent methods for text categorization and stylometric applications are based on word embeddings and deep learning [25, 26, 236]. This section describes the first notion. The main idea behind word embeddings [234, 263, 264] is to represent each word-type as a vector generated according to word semantics. In this view, each word corresponds to a point in a vector-space $R^q$, where $q$ indicates the dimension number of the representation space (usually $q = 100$ to 300). Such a representation is built without considering a dictionary, a thesaurus, or other linguistic databases but extracted from the text sample itself. Such an approach is called *unsupervised* because the target category is not provided. The resulting lexical representation can then be used in other applications such as in deep learning models or for solving the authorship attribution question.

For example, a word embeddings representation could have been generated for the words *author*, *book*, and *cats* with a size $q = 3$ as follows:

$$author = \begin{bmatrix} 0.68 \\ -0.51 \\ 0.02 \end{bmatrix} \quad book = \begin{bmatrix} 0.81 \\ -0.75 \\ 0.03 \end{bmatrix} \quad cats = \begin{bmatrix} -0.75 \\ 0.41 \\ 0.91 \end{bmatrix}$$

There is no clear interpretation for each component of these vectors. This is a property of the distributed representation; there is no one-to-one correspondence between a specific component of the vector and a concept. Only the entire vector is useful to represent the target entity (e.g., cats).

However, when two words own a similar meaning, their representations are closer. Using a similarity function, one can then compute the proximity (or distance) between two vectors (or two words). For example, the dot product (see Sect. 6.1) fulfills this requirement. With our data, the dot product between *author* and *book* is computed as: $0.68 \times 0.81 + (-0.51) \times (-0.75) + 0.02 \times 0.03 = 0.934$, while the dot product between *book* and *cats* returns -0.888 (or $-0.701$ between *author* and *cats*). Clearly the concepts of *author* and *book* are closer or more similar than the semantics relationship *book* and *cats*.

This scheme also allows us to *compute* the representation of related concepts. For example, as each word-type is a vector, one can derive the vector *queen* (the semantic representation of the concept *queen*) as vector(*king*) – vector(*man*) + vector(*woman*) = vector(*queen*) [264]. In a similar way, semantic relationships between pairs of words can be discovered, for example, *Paris* is to *France*, what *Rome* is to *Italy*, or vector(*France*) + vector(*capital*) = vector(*Paris*). Sometimes the compositionality of the language could be verified, for example, vector(*airlines*)

+ vector(*Germany*) = vector(*Lufthansa*). Even morphological relationships could be identified such as vector(*children*) – vector(*child*) + vector(*cat*) = vector(*cats*). In addition, one can generate a distributional thesaurus based on word embedding models to regroup under an entry related terms (e.g., under the word *cat*, one can find {kitty, pussycat, poodle, pet, meow, sweetie, Garfield, to meow, baba, feline, animal, etc.}) [117].

This word representation is built from the context of each word occurrence under the assumption that similar contexts imply similar representations and similar meanings. Thus, when two words share many similar contexts over many sentences, their meaning must be similar or, at least, related. This hypothesis is behind the concept of distributional semantics justified by linguistics theories: "You shall know a word by the company it keeps" [119] or "The meaning of a word is its use in the language" [421]. Moreover, a larger distance within a sentence between words usually implies that their relationship must be weaker [152].

Therefore, a pair of vectors depicting a small distance between them could be viewed as synonyms. However, it was found that two antonyms could also present a small distance because they often share similar contexts, for example, the context of the prepositions *over* or *under* (e.g., "the plane flew over/under the bridge," "the value is over/under the limit"). Of course, linguists differentiate word associations using various semantic relationships (e.g., synonymy, hyponymy (e.g., football is a kind of sport), meronymy (e.g., motor is a part of a car), polysemy (jaguar as an animal or a car), etc.). Of course, these labels are absent in a word embeddings representation.

Moreover, the compositional aspect in a language is limited. For example, the meaning of collocations cannot be inferred by simply adding two vectors in expressions like *red book* or *cat fish*.[6] The presence of homographs could also hurt the overall quality of the representation (e.g., the same vector *saw* could be used for the two occurrences in the sentence "I saw a man with a saw").

To generate a word vector representation, a large sample of text must be available. For example, the Word2Vec proposed by Mikolov [263] is based on news articles (the Google dataset containing 100 billion tokens). Other corpora have been used such as Wikipedia articles, tweets, or a crawl of web pages (e.g., GloVe [300]). The important factor is to generate the model with a huge amount of textual data and to ignore rare events. Thus, some preprocessing can be applied to the corpus before its use, for example, to ignore terms that have an absolute frequency of four or smaller [263].

To generate a word embeddings framework, several neural network models have been suggested. For example, having $m$ distinct word-types, one can represent each of them by a vector of $m$ binary values, all being equal to 0 except the $i$th containing a 1 to indicate the target word (one-hot encoding). In this case, $m$ also indicates the

[6] A similar problem occurs in English with verbs with particles such as to *give up* or to *pick up*. Moreover, the distance between these two elements could be larger than 5 (a default value) such as in "Please, turn the lights in the first floor near to the kitchen off."

vocabulary size. As an example, Fig. 7.8 depicts an overview of a neural network to generate a word embedding representation with, as input, the word can be stored in one-hot encoding form (with $m = 10,000$).

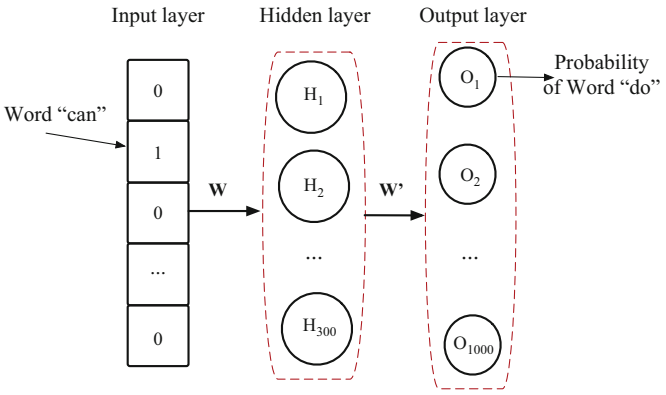


**Fig. 7.8** Neural network used to generate a word embeddings representation

In this architecture, all neurons of the input layer are associated with all neurons of the hidden layer (fully connected network) and similarly between the second and third layer. Some weights $w_{k,j}$ could be very small or even equal to zero. Instead of representing all the links between the input and hidden layer, the connection pattern is indicated by a matrix denoted W with 10,000 rows and 300 columns.[7] Similarly, and as shown in Fig. 7.8, a matrix W′ with 300 rows and 10,000 columns specifies the connections between the hidden and output layer.

In the output layer, the proposed solution corresponds to the activation values computed by the *tanh* function. As each of these values appears in the range $[-1, 1]$, the answer is not a probability distribution. To solve this difficulty, the *softmax* operator is applied to transform the output vector into a probabilistic distribution over the 10,000 words (see Eq. 7.8). After this transformation, and according to the input word, one can inspect the probability estimations for the entire vocabulary.

As the learning stage must determine for the matrix W (containing $10,000 \times 300 = 3$ million values), such a volume clearly explains why the learning sample must be huge.

Based on this general neural network architecture, different word embedding models are possible. As an example, the generation of the Word2Vec approach [263] could be achieved according to two possible models. The first one predicts the surrounding words of a given one (skip-gram model), while the second determines

[7] With the matrix notation, the input vector is transposed before applying the multiplication with W producing a new vector $1 \times 300$ that is then multiplied by W′ (300 rows, 10,000 columns) resulting in the output vector $1 \times 10,000$.

the target word given a context (cbow, continuous bag-of-words). Denoting the word-type in the position $t$ by $w_t$, the skip-gram model needs as input $w_t$ and returns its most probable nearby context $(w_{t-b}, w_{t-b-1}, \ldots, w_{t-1}, w_{t+1}, \ldots w_{t+b})$ in which $b$ is the window size (e.g., by default $b = 5$). To put it another way, we want to learn $p(context|w_t)$ for different words appearing in the context. The cbow model is used to predict the occurrence of the $w_t$ given a context $w_{t-b}, w_{t-b-1}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+b}$ or to learn $p(w_t|context)$.

The word embedding representation is learned according to an objective function that maximizes the probability of any context given $w_t$, for all word $t = 1, 2, \cdots m$ as depicted in Eq. 7.11 (maximizing the likelihood).

$$Max \prod_{t=1}^{m} \prod_{j \neq 0, j=-b}^{b} p_v\left(w_{t+j}|w_t\right) \tag{7.11}$$

This formulation is usually transformed into the negative of the log likelihood as:

$$Min -\frac{1}{m} \sum_{t=1}^{m} \sum_{j \neq t, j=-b}^{t+b} \log p_v\left(w_{t+j}|w_t\right) \tag{7.12}$$

where $w_j$ indicates the word-type occurring at position $j$, $w_t$ the input word-type, $b$ is the window size, and $p_v()$ is the neural network classifier (probability) that the word $w_j$ appears in the context of word $w_t$, based on the representation $v$. The learning process will maximize this probability for all possible words $w_t$, (in the context of size $b$). For example, when the sequence "yes we can do it" appears in the training corpus with the target word *can*, the following word pairs are used to learn the context of *can* {(yes, can), (we, can), (do, can), (it, can)}. Finally, the word embeddings representation for a given input word corresponds to the values stored in the hidden layer (see Fig. 7.8).

Of course, various hyperparameters must also be fixed such as the window size ($b = 5$), the number of dimensions used in the representation ($q = 300$ in Fig. 7.8), the minimum number of occurrences of five (or more) terms to be taken into consideration, the number of iterations in the learning stage (or number of epochs), the learning rate (e.g., $\alpha = 0.005$), and other parameters related to the underlying neural network structure. To minimize the training time (number of epochs, number of word-tokens, vocabulary length, window size, size of the hidden layer), efficient learning solutions have been proposed [263, 322]. The overall effectiveness of a word embedding representation is related to certain system design choices and hyperparameter optimizations [240].

To propose an authorship attribution model based on the word embedding representation, each document D is stored as a vector composed by $m$ selected words (or stylistic features). Each of them also corresponds to a vector (of size $q$) and the document D is simply the weighted average of these components as shown

in Eq. 7.13.

$$D = \sum_{i=1}^{m} r_i \cdot \begin{bmatrix} v_{i,1} \\ v_{i,.} \\ v_{i,q} \end{bmatrix} = r_1 \cdot \begin{bmatrix} v_{1,1} \\ v_{1,.} \\ v_{1,q} \end{bmatrix} + r_2 \cdot \begin{bmatrix} v_{2,1} \\ v_{2,.} \\ v_{2,q} \end{bmatrix} \cdots + r_m \cdot \begin{bmatrix} v_{m,1} \\ v_{m,.} \\ v_{m,q} \end{bmatrix} \tag{7.13}$$

where $r_i$ indicates the weight associated with the $i$th word-type. As a possible implementation, each $r_i$ corresponds to the relative frequency of the word-type in document D (or author profile), see [210].

With this model, the set of the $m$ word-types included in each document representation must be specified. To define them, previous stylistic studies have shown that most frequent words [46, 339] or functional words [431] provide effective stylistic features. Unlike previous authorship models, the word embedding representation takes account of the context of those selected terms. Therefore, it is not only the differences in frequencies (or Z score in the Delta model [46]) but the differences in the context that is taken into account. For example, when an author tends to use constructions such as *of the* more frequently, the word embedding vector of terms of and the reflects this lexical usage.

A preliminary evaluation of the use of word embeddings for authorship attribution [210] indicates that the overall effectiveness is comparable to the state-of-the-art models. As with many other machine-learning based approaches, the system needs training data having similar characteristics to that of the test data (e.g., extracted from the same time period, written in the same text genre and register, and when possible having similar topics). This conclusion was also reached with other NLP tasks, for example, to generate a distributional thesaurus in which the performance levels are still low even with word embedding approaches [117].

Even if the deep learning model requires the specification of different parameters, the proposed default values tend to produce high performance levels. Small variations around the default values do not significantly modify the achieved effectiveness. Finally, the decision proposed by a word embeddings model must be justified and some degree of support or belief that the proposed author is the true one must be given. These two aspects are more difficult to specify concretely and could be a subject of future work in the field. Thus, one can always view the proposed attribution as a complementary one that other authorship attribution models can confirm to achieve a higher degree of confidence about the final attribution.

## 7.8 Deep Learning and Long Short-Term Memory (LSTM)

The term *deep* is associated with neural networks built with more than one single hidden layer (see Sect. 7.6). If the principles stay the same for deep neural networks (DNN), the huge number of parameters to be learned (the weights and biases) require larger learning samples, more time, and need more resources. Moreover,

the number of layers could lead to a vanishing error rate and therefore the needed modifications cannot be performed to distant layers.

During the last three decades, various studies have proposed efficient solutions to those problems. First, the recent progress in hardware has produced more efficient CPUs and larger cache memories, allowing a better efficiency in the numerous computations required by deep learning models. For example, as deep learning models require many matrix multiplications, GPU (graphical processing unit) was specifically built for that purpose and thus has been used now for this new intent (currently NVIDIA produces such devices directly for deep learning applications). Moreover, dedicated devices and hardware implementation in chips have been proposed [39]. Second, various improvements in the algorithms have sped up the learning stage. For example, instead of modifying the weights after each instance, batch update has been suggested as an effective strategy. Some random effects have been inserted during the learning stage to reduce the risk of overfitting. Third, large databases have been made available to allow learning with a huge number of instances (e.g., newspaper archives, Wikipedia, social networks for text data, or FlickR for photos, etc.).

As various deep learning architectures [53, 134] (e.g., BERT [255]) could be applied for different stylometric applications, this section describes the main ideas behind recurrent neural networks and, more specifically, the long short-term memory (LSTM) model [158]. In this view, the first important concept is to build a neural network able to temporarily store information. A quick look at Fig. 7.7 or Fig. 7.8 shows that these feed-forward neural networks cannot store information or their intern state for long. In the current case, the concept of *long term* means able to establish the agreement between a subject and a verb in a sentence such as "The *book* that John bought this morning *is* red."

Through activation, the flow goes from the input layer to the output one. No information can be stored for later use. To achieve this, recurrent neural networks (RNN) have been proposed [25, 132, 329], usually to process sequential data (e.g., flow of numbers over time, sequence of words of a given text). A very simple example is depicted in Fig. 7.9.

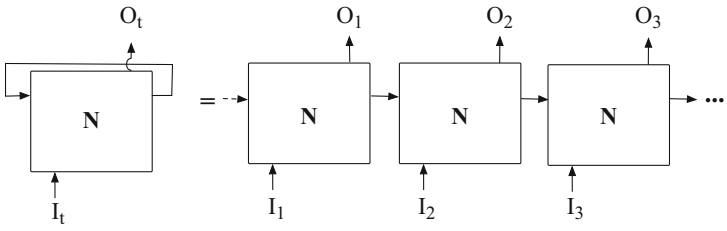


**Fig. 7.9** Left: A simple RNN. Right: The unrolled simple RNN

On the left, the neuron N receives an input vector denoted $I_t$ and produces an output vector indicated by $O_t$. However, a second exit appears on the right returning

the value into the neuron on the left. With this additional link, a simple recurrent unit is generated, able to store a vector to be processed in the next step. One can view this as a state vector or a cell state storing the information seen and processed in the past. With this solution, past information can be used to derive more appropriate answers in the future.

But why consider such a state vector? In some applications, the input values are coming in a sequence such as in speech or more generally in the language. In fact, a text is a series of sentences and each of them corresponds to a sequence of words (and even a sequence of characters [200]). The state vector can memorize information useful for producing the right output later. For example, in language generation [385], the fact that the previous noun is feminine implies that the pronoun must be *her* instead of *him* or *it*. When considering a larger part of the history of past items and their processing, one can discover topic change [5] or, in our context, to identify the true author or some demographics information about him.

In RNN, the input corresponds to both a new item in the sequence denoted $I_t$ and the cell state derived from previous computations. On the right part of Fig. 7.9, the RNN is unrolled and the sequence of the input data is more clearly provided as $I_1, I_2, I_3, \ldots$ as well as the corresponding output $O_1, O_2, O_3, \ldots$ For example, the input sequence might correspond to the word sequence in a source language, while the output sequence could be the same sentence in a target language. In a stylometric application, the output vector $O_t$ could indicate the probability distribution over the possible categories.

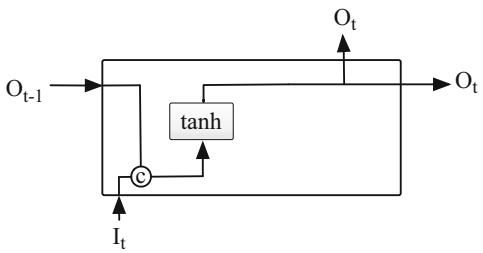


**Fig. 7.10** A simple RNN unit

The intern structure of the neuron N and the overall network layout varies from one RNN model to the next. To illustrate this, Fig. 7.10 describes a simple RNN unit. In this model, the state vector and the input vector $I_t$ are concatenated (a "c" appears inside the circle joining the two vector paths). This enlarged vector forms the input of a neural network layer using *tanh* as activation function. This layer is symbolized by a box in Fig. 7.10. Finally, the resulting vector is then sent to the output and a copy forms the new state vector.

The intern structure of each long short-term memory (LSTM) unit [158] is more complex as shown in Fig. 7.11. First, each unit has two entries for storing information derived from past items and to be delivered to the next unit. A more

complete description of the intern processing performed by each LSTM unit can be found in [287].

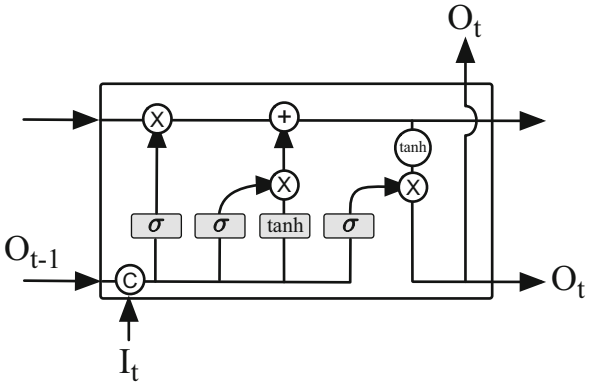


**Fig. 7.11** A long short-term memory (LSTM) cell

In a few words, the top vector path represents the *cell state* or *state vector* that is first multiplied with a second vector and then added with a third one before joining the next unit. This path conveys the cell state corresponding to the long-term memory from one unit to the next.

On the bottom, the second entry to the left corresponds to the output of the previous unit (symbolized with $O_{t-1}$) or the short-term memory. This output is concatenated with the next input item $I_t$ to form the working vector. Four copies of this vector are processed by three layers using a sigmoid activation function (denoted with the symbol $\sigma$) and by one neural layer using $tanh$ as activation function.

The main computation of the LSTM unit is performed with this working vector. Starting from the left, the first sigmoid layer (called the *forget gate* layer) returns a vector with values between 0 and 1. Then this vector appears in an element-wise (or pointwise) multiplication with the cell state. One can observe this as a way to weight the components of the cell state coming from the previous unit, amplifying some of them, reducing others. In other words, the forget gate must decide which components of the state vector are still important and which ones can be reduced or even ignored.

In the middle, one copy of the working vector goes through a sigmoid layer employed to specify which components must be updated and the amplitude of this modification. This part with the next $tanh$ layer forms together the *input gate*. The other copy of the vector crosses the $tanh$ neural layer to define the current candidate values to be included in the cell vector and to produce the output of this unit. An element-wise multiplication combines these two vectors that are then added to the cell state on the top. This modified cell state goes directly to the next LSTM unit on

the one hand, and, on the other, a copy goes through the second $tanh$ layer to be the first part of the current output vector.

On the right, the working vector crosses the third sigmoid layer (to select the components useful to generate the current output vector). After this neural layer, the vector is element-wise multiplied with the transformed cell state (that just goes through the $tanh$ function) to generate the output vector $O_t$ that is also transferred to the next LSTM unit.

Different variants of this basic LSTM unit have been proposed, usually by adding additional paths between the cell state and the different neural layers (e.g., [128]). However, few experiments have been conducted with this model [350] or with other deep learning approaches in the stylometric domain. Currently their overall best effectiveness levels could be close to those achieved by the best systems, but not clearly higher. For example, during the last three CLEF PAN evaluation campaigns in 2017 [309], 2018 [315], and 2019 [314], the best performing approaches were not based on deep learning strategy. Of course, this conclusion could change in the near future with the development of new deep learning models and network architectures. However, together with high effectiveness, a useful stylometric approach must provide some precise linguistic explanation justifying the proposed decision. This aspect is not currently achieved with deep learning models.

## 7.9 Adversarial Stylometry and Obfuscation

Until now, the main focus has been put on identifying the author name or some of his demographics. However, one can see the other side of the coin with the objective to protect an author's anonymity or simply to develop techniques to hide some of his demographic information. In this perspective, adversarial stylometry is defined as designing and implementing techniques to fool the outcome of stylometric analyses. From this broad view, different contexts and applications may appear.

**Imitation** The first possible use of adversarial stylometry consists of imitating the style of a given author (e.g., Jane Austen) or to pastiche a well-known politician (e.g., see Fig. 5.1 showing a tweet that could have been sent by Trump). The idea is therefore to write a text to closely reflect the authorial voice of a target person including some of his lexical or syntactic idiosyncrasies. To have an idea about the effort required to achieve this goal, one can first consider some manual imitation experiments.

As an example, the Brennan–Greenstadt corpus [188] contains 15 text samples (around 5000 words) written by 15 authors. In addition, those writers produced another set of 15 texts (around 500 words) in a way that hides their own identity. Moreover, a second set of 15 texts (around 500 words) have been generated with the goal to imitate McCarthy's style (in the novel *The Road* (2006)). To reveal the true identity behind those obfuscated texts, different authorship attribution methods (distance-based, KLD, SVM, naïve Bayes, etc.) have been applied. The feature set

could be words, letters, character $n$-grams, syllables, with or without the punctuation signs, etc. These methods were usually unable to detect the true author behind a manually obfuscated text (maximum accuracy being 42%). In the imitation stage, the texts were often attributed to McCarthy. One can conclude that a manually obfuscated text is harder to detect. One must, however, note that the texts were short (500 words) with a limited number of features (15) used by the authorship systems. This corpus has since been extended to 45 writers using Amazon Mechanical Turk [42] and the results replicated, confirming the fact that human beings can control their own writing style and adopt stylistic features of another, at least for producing short texts.

In a related study based on the *Federalist Papers*, Kacmarcik and Gamon [196] have demonstrated that modifying Madison's style to get Hamilton's is possible when modifying the frequencies of fourteen terms (e.g., *on*, *powers*, *there*, *to*, *upon*, etc., see Appendix A.3). To test the effectiveness of the proposed method, a linear SVM was generated to learn and detect the fingerprint of both authors. After modifying the text surrogates, the system recognized Hamilton's style instead of Madison's. In this example, it is clear that both the style of the original and target author must be analyzed and represented. Their differences must then be identified to know in which directions the modifications must be performed.

Modifying the text itself is a more complex task and consequently implies the alteration of the frequencies of other terms. But this second experiment indicates that manually imitating the style of a given author cannot be easily detected by a stylometric analysis.

Except for short texts, no automatic solution has been proposed to modify a text to both imitate the style of a given author and counterfeit stylometric models. As a more complex application, the objective could be to change one (or more) demographic dimensions for another value. For example, to modify a text written by a woman in order to appear as if written by a man or more complex from a female teenager to an adult male.

**Obfuscation** Under this expression, the objective is to erase the stylistic idiosyncrasies corresponding to a given individual author or to remove features related to some of his demographics. As for the imitation, the aim is always to defeat a stylometric investigation but in the current case without specifying the style to be imitated. For privacy and security reasons, it could be justified to protect the real identity of the writer (e.g., because he is a whistle-blower, an artist, or journalist). For example, a well-known author might want to preserve his anonymity or use a penname to write with a renewed style (e.g., the Gary–Ajar's case [295]) or to script a new text genre (e.g., J. K. Rowling with the crime novel *The Cuckoo's Calling* (published in 2013) [191]). In science fiction, it is not unusual to see an author writing under several pseudonyms, for example, Randall Garret also known as David Gordon, Darrel T. Langart, John Gordon, Richard Greer, etc.

Let us start with the first case. Can the computer be programmed to automatically remove the stylistic items that can identify the true author? This objective was submitted to the CLEF PAN evaluation campaign in 2016 [308], 2017 [309], and

2018 [310]. The problem to be solved is stated as follows: given two documents by the same author, paraphrase the designated one so that the author cannot be verified anymore.

As a first solution, one can simply use freely available translation tools to convert the original text into a target language and then back to the source language. The idea is to generate a new version of the original text by replacing some words or expressions by their synonyms and by varying the sentence structure. This approach was first suggested by [316] using English as the source language and French as the target one. Thus, a single step translation process was performed, translating the original English text into French, then back to English. In [42] and [4], this experiment was repeated with German or Japanese as the target language. Moreover, instead of considering only a single step translation, a two-step translation process has been applied (e.g., English → German → Japanese → English). Short sentences are easier to translate than longer ones but usually lack obfuscation. With complex sentences, the translation task is harder and the meaning is not always preserved. After a single or a two-step automatic translation stage, the authorial voice included in the original text was not recognized anymore by three different stylometric models [42]. During CLEF PAN evaluation campaigns [308], circular translation was also proposed by some participants with moderate success.

However, other approaches have been suggested [144, 308–310]. Usually, the first stage is to generate a representation of the style to be obfuscated, usually based on words and some additional lexical or syntactical measurements (e.g., mean word length, percentage of big words, mean sentence length, etc.). Even if letter $n$-grams might provide a more efficient approach, it is not clear how to alter the frequencies of $n$-grams of letters (e.g., how to reduce the number of bigrams "en" and to increase the frequency of "tio").

In a second stage, the main flow is to disturb the frequency of the selected stylistic features (e.g., functional terms, isolated words, or average lengths). In this goal, one can replace a word by one of its synonyms less used in the original text. Based on a thesaurus (e.g., WordNet [113, 114]), one can consider replacing a large number of words or expressions with their possible synonyms.

Some grammatical constructions could also be modified (e.g., expanding contractions (I'll → I will) or the reverse) as well as combining or splitting sentences. One can also remove some connectives (e.g., *moreover*, *but*, *and*, *however* appearing at the beginning of a sentence) or passages written between parentheses. As another technique, one can add a few spelling errors (or correct some of them) [271]. Applying the aforementioned techniques usually tends to obfuscate the text and reduce significantly the performance of 44 author verification approaches developed during several CLEF PAN campaigns [144, 308–310].

Does that mean that the problem is solved? The answer seems positive because these strategies could circumvent their opponents formed by several authorship verification systems. Obfuscated texts must however be evaluated based on three orthogonal dimensions, namely their safety, soundness, and sensibility [144, 310]. The safety aspect reflects the ability of an obfuscation system to generate paraphrases that effectively hide the author's identity, meaning that author verification

systems are no longer able to determine the true author's identity. This aspect was usually achieved by the single- or two-step translation strategy.

But two additional constraints must be respected. The soundness focuses on the ability of an obfuscation approach to retain the semantics of the original text, meaning that the obfuscated text must have a similar meaning. Third, the sensibleness reflects the ability of an obfuscation approach to produce readable and grammatically correct paraphrases. Whereas the safety aspect can be evaluated automatically (e.g., using various verification methods when applied to the original and obfuscated text), the soundness and sensibility checks require a manual evaluation by human assessors.

Based on CLEF PAN results, adopting a conservative obfuscation system (at most one word is replaced per sentence), the soundness and sensibleness are usually preserved, not the safety. With more aggressive approaches, the safety is achieved but the resulting obfuscated texts are hard to read (sensibleness) and the text meaning has been changed [144, 310].

# Part III
# Cases Studies

The third part of this book is dedicated to real applications in stylometry. The main purpose is to present to the reader how one can apply the described methods with real data. A direct application of the previously described methods is not always the best approach. From time to time, some adaptations are required. Such variations could be limited to ignore one author, to apply the suggested method with another set of features, or that the applied method could solve another problem. In addition, the final decision is usually taken based on a combination of evidences, each of them being supported by a given stylometric method. This is the usual case when such a decision has an important consequence.

Chapter 8 exposes the central question of the stylometric domain: How can one identify the true author of a text? In this case, the question is to identify who is the secret hand behind the *nom de plume* Elena Ferrante, a worldwide-known Italian writer, author of the *My Brilliant Friend* saga (four novels written between 2011 and 2014). Even though the 150 novels were written in Italian, several authorship attribution methods described in the previous chapters have been applied on this corpus of novels authored by 40 distinct writers. As it will be shown, this authorship attribution problem demonstrates a consistent finding; the same conclusion is achieved by all chosen approaches using different text representations and stylistic markers.

Chapter 9 describes two problems related to the general context of author profiling. Moreover, instead of considering literary works, a document is defined as a set of 100 tweets, written in English by either a bot, a man, or a woman. Extracted from the CLEF PAN 2019 evaluation campaign, the corpus contains 6760 such documents. The first problem is to design a text classifier able to discriminate between tweets generated by a bot and those written by a human. This could be possible when the computer is able to discover which stylistic indicators can be strongly associated to either a bot or to a human being. The second problem is to determine whether a set of tweets have been scripted by a man or a woman. To achieve this, the system must be built on fingerprints that differ according to the author's gender. As explained in this chapter, both problems can be solved by a computer but without achieving a 100% accuracy rate.

Chapter 10 is focusing on another text genre and domain. In this case, the political field has been selected with a corpus composed of the US presidential speeches from G. Washington (1789) to D. Trump (2020). In this chapter, several techniques are applied to determine the stylistic evolution of the governmental speeches over more than 230 years. Besides the style of most frequent words used in a given language or time period, one can apply techniques extracting the vocabulary characterizing a given presidency. The variation of those terms over the years can also be useful to detect some writing trends. Moreover, the identification of the distinctive words allows the system to identify typical sentences reflecting a given president. In addition, new methods based on wordlists are presented to determine some general characteristics of the rhetoric and writing style of more recent US presidents.

# Chapter 8
# Elena Ferrante: A Case Study in Authorship Attribution


To illustrate the application of four authorship attribution methods described previously, this chapter presents a real case in authorship attribution. With the translation of the successful novel *L'amica geniale* (2011) (*My Brilliant Friend*, in 2012) into many languages, the *nom de plume* Elena Ferrante has gained worldwide attention. Before, this name was already known in Italy with the success of her first two novels (*L'amore molesto*, 1992 (Troubling Love, 2006) and *I giorni dell'abbandono*, 2002 (The Days of Abandonment, 2005)). She obtained a real international reputation with the successful tetralogy of the *My Brilliant Friend*'s saga (*L'amica geniale* (2011), *Storia del nuovo cognome* (2012), *Storia di chi fugge e di chi resta* (2013), and *Storia della bambina perduta*[1] (2014)), a story brought to the screen by the RAI and HBO channels.

But the real identity of E. Ferrante is still a mystery. In Italy in particular, several names have been proposed: mainly well-known female novelists originating from Naples (e.g., Milone, Parrella, Ramodino), but also some men (e.g., De Luca, Piccolo, Prisco, etc.), and even essayists (e.g., G. Fofi), academics (e.g., M. Marmo), critics, journalists (e.g., D. Bignardi), film directors (e.g., M. Martone), translators (A. Raja), and even publishers (e.g., S. Ozzola). These suggestions have been formulated by columnists or literary scholars with some intuition about stylistic similarities, and, in Raja's case, according to royalties received.

Let us look more carefully at some of Ferrante's dimensions. First, one can suspect a female writer is behind, for example, *My Brilliant Friend*. This story portrays the life of two Neapolitan girls and their relationships with their families and friends from a very feminine point of view. It could be assumed that the dialogues, the emotions, and sentimental descriptions could not be written like that by a man. Thus, Ferrante must be a woman. As this saga is located in Naples and the descriptions so closely reflect the reality, this must imply that the author came

[1] In English with *My Brilliant Friend* (2012), *The Story of a New Name* (2013), *Those Who Leave and Those Who Stay* (2014), and *The Story of the Lost Child* (2015).



J. Savoy, *Machine Learning Methods for Stylometry*,
https://doi.org/10.1007/978-3-030-53360-1_8

from Campania or even that s/he must have been born in Naples. In addition, the target novelist cannot be a young writer, but had to have lived in the fifties to know all the daily details included about that period.

Despite this success story, the scholars and literature experts did not really study her style and analyze her novels. Therefore, in 2017 professors Arjuna Tuzzi and Michele A. Cortelazzo decided to collect a large corpus of contemporary Italian writers to draw Elena Ferrate's profile [408] and to try to identify the probable real author behind this pseudonym. Section 8.1 exposes the main features of this corpus generated at Padova University (for more details, see [407, 408]).

To determine the true author of a novel, numerous authorship attribution methods have been proposed (e.g., Juola and Vescoci [188] suggest more than 1000 approaches). Therefore, it may be hard to believe that a single attribution model could always provide the correct answer in all circumstances. Therefore, to ascertain a proposition with a higher degree of certainty, several approaches must be taken into account. Such an evaluation methodology has been suggested by Juola [193] and [347]. To be accepted in a US court [54], such methods must reflect the state of the art in the domain. They must have demonstrated their effectiveness and robustness in several contexts using different test collections. To respect these constraints, Sect. 8.2 exposes the application of Principal Component Analysis (PCA) to analyze the authors' profiles with a reduced set of the most frequent word-types (MFWs). To complement this, the stylometric analysis will be based on the Delta model comparing each of Ferrante's novels with 39 possible Italian writers in Sect. 8.3. When applying Labbé's intertextual distance in Sect. 8.4, close to all of the vocabulary will be employed to identify the real author behind Ferrante's books. When presenting the Zeta test in Sect. 8.5, only a selected fraction of the vocabulary is used to find the possible relationship between Ferrante and Starnone. With this structure, the selected stylometric approaches are working at different granularity levels, starting with the entire works of each author (Sect. 8.2), continuing at the book level (Sects. 8.3 and 8.4), and finishing with a set of chosen words (Sect. 8.5). Section 8.6 presents a qualitative analysis that completes this first application of stylometric models. Finally, a conclusion exposes the main findings and remaining open questions of this real application.

## 8.1 Corpus and Objectives

As for all authorship attribution problems, data must be collected with the constraints that the analyzed documents must correspond to the same text genre and be written during the same period as the one in question. To achieve high accuracy, each text must be relatively long (more than 5000 words) and of high quality (and thus we must reject poor OCR recognition). Of course, all selected books will appear in their Italian version (no translation that could hurt the stylistic features left by the true author, see [332]).

Our collection was generated by a team of researchers at Padova University, under the supervision of Prof. Arjuna Tuzzi and Prof. Michelle Cortelazzo [408]. Their selection of literary works was based on the following additional considerations. As it is assumed that Ferrante could be a woman coming from Campania, more novels are included that are written by novelists coming from Naples and the surrounding region. Moreover, a bias for books authored by women was applied. To respect the text genre constraint, only novels written for adult readers[2] have been selected. Thus, they ignored a children's story authored by Ferrante (*La spiaggia di notte*[3] (2007)) and a collection of letters, interviews, and essays (*La frantumaglia* (2016)).

To limit the number of contemporary Italian novels to be included in the corpus, the Padova team selected only best-sellers or award-winning novels or those praised by the literary critics. Finally, when establishing the list of possible authors, they tried to select all names mentioned as the possible novelist of Ferrante's works.

The final list is reported in Table 8.1 together with their gender, the region, and the number of novels included in the corpus. As shown below, this corpus contains 150 novels written by 40 different authors (27 men, 12 women, and Ferrante). Each author is represented by at least two novels and as many as ten in Starnone's case. Ferrante is included, with seven books (including the four novels of her tetralogy *My Brilliant Friend*). To respect the possible geographical origin of Ferrante, ten authors come from Campania (Naples) (namely De Luca, De Silva, Milone, Montesano, Parrella, Piccollo, Prisco, Ramondino, Rea, and Starnone). This regional aspect is important in Italian, due to the presence of spelling differences between regions (diatopic variation) and the use of dialect-specific words and expressions.

Finally, a careful editing process has been undertaken to remove all elements not belonging to the text itself (e.g., page numbers, running titles, etc.), as well as a thorough checking of the spelling.

Even if a real effort has been done to embrace all possible writers, it is always possible that an unknown person wrote all of the Ferrante books and nothing else. As another hypothesis, this unknown author might have written other text genre stories not considered when generating this corpus. These reasons explain why the names of M. Marmo or A. Raja do not appear in the final list. The first one is a professor in contemporary history in Naples, and the second a translator of German books into Italian. Neither wrote any novels for adult readers.

To define the regions, Table 8.1 utilizes the following abbreviations: *Camp*. for Campania (Napoli), *Lomb*. for Lombardia (Milano), *Piem*. for Piemonte (Torino), *Emil*. for Emilia Romagna (Bologna), *Sard*. for Sardegna (Nuoro), *Tosc*. for Toscana (Firenze), *Sici*. for Sicilia (Palermo), and in full names: Lazio (Roma), Puglia (Bari), Friuli for Venezia-Giulia (Trieste), and Veneto (Venezia). More information about

[2]One can object that this text genre is too broad because novels for adult readers could include different categories such as romance, adventure, historical, thriller, fantasy, etc.

[3]This book appeared under the title *The Beach at Night* (2016).

**Table 8.1** Author name, gender (M/F), region, and the number of novels

| Name | Gender | Region | Number | Name | Gender | Region | Number |
|---|---|---|---|---|---|---|---|
| Affinati | M | Lazio | 2 | Montesano | M | Camp. | 2 |
| Ammaniti | M | Lazio | 4 | Morazzoni | F | Lomb. | 2 |
| Bajani | M | Lazio | 3 | Murgia | F | Sard. | 5 |
| Balzano | M | Lomb. | 2 | Nesi | M | Tosc. | 3 |
| Baricco | M | Piem. | 4 | Nori | M | Emil. | 3 |
| Benni | M | Emil. | 3 | Parrella | F | Camp. | 2 |
| Brizzi | M | Emil. | 3 | Piccolo | M | Camp. | 7 |
| Carofiglio | M | Puglia | 9 | Pincio | M | Lazio | 3 |
| Covacich | M | Friuli | 2 | Prisco | M | Camp. | 2 |
| De Luca | M | Camp. | 4 | Raimo | M | Lazio | 2 |
| De Silva | M | Camp. | 5 | Ramondino | F | Camp. | 2 |
| Faletti | M | Piem. | 5 | Rea | M | Camp. | 3 |
| Ferrante | | | 7 | Scarpa | M | Veneto | 4 |
| Fois | M | Sard. | 3 | Sereni | F | Lazio | 6 |
| Giordano | M | Piem. | 3 | Starnone | M | Camp. | 10 |
| Lagioia | M | Puglia | 3 | Tamaro | F | Friuli | 5 |
| Maraini | F | Tosc. | 5 | Valerio | F | Lazio | 3 |
| Mazzantini | F | Lazio | 4 | Vasta | M | Sici. | 2 |
| Mazzucco | F | Lazio | 5 | Veronesi | M | Tosc. | 4 |
| Milone | F | Camp. | 2 | Vinci | F | Lomb. | 2 |

this corpus can be found in [407] or in the proceedings of a workshop on this topic [406].

In total, the corpus contains 9,821,883 word-tokens (for 156,815 distinct word-types or 83,326 lemmas), ignoring numbers and punctuation symbols. In average, each novel contains 64,062 tokens (standard deviation: 38,228). The largest book is composed of 196,914 tokens (Faletti, *Io uccito*, 2002) and the smallest of 7694 tokens (written by Parrella, *Behave*, 2011, the only work with fewer than 10,000 word-tokens). For Ferrante's novels, the average size is 88,933 word-tokens (min: 36,222 (*La figlia oscura*), max: 136,945 (*Storia della bambina perduta*)). In total, Ferrante's writings represent 6.48% of the corpus, while those of Faletti constitute the largest share (6.6%) followed by Starnone (6.4%) and Mazzucco (6.15%). The smallest contribution is provided by Parrella (0.36%), followed by Vinci (0.58%) and Nori (0.64%).

These novels have mostly been published between 1987 and 2016. However, for three authors (namely Maraini, Morazzoni, and Prisco), one novel appears before 1987 in order to have at least two novels per author or to collect a reasonable set of works for each of them. For Prisco, this choice was also supported by the fact that his name was proposed as the possible novelist behind Ferrante's books.

As all these books are written in Italian, it could be useful to point out some linguistics differences with the English language. Focusing on the most frequent

words, Italian has more forms to represent the determiners *the* (e.g., *il*, *la*, *lo*, *l*, *i*, *le*, *gli*) or *a/an* (e.g., *un*, *una*, *uno*). If we need to translate the expression *of the* in Italian, different possible words can be used depending on the contexts and on the gender (masculine or feminine) and number (singular or plural) of the following noun (e.g., *del*, *dello*, *della*, *dell*, *dei*, *delle*, *degli*, *dal*, *dallo*, *dalla*, *dall*, *dai*, *dalle*, *degli*). A larger variability is present when considering the various forms of the verb *to be* (e.g., present: *am*, *are*, *is*, vs. *sono*, *sei*, *è*, *siamo*, *siete*) or to have (e.g., present: *have*, *has* vs. *ho*, *hai*, *ha*, *abbiamo*, *avete*, *hanno*).[4]

Finally, as preprocessing, and for all experiments, each text has been analyzed by the TreeTagger POS tagger[5] to derive both the word-tokens (tokenization) and the lemmas (dictionary entries). When the lemma cannot be defined by the tagger, the corresponding token is used (usually when dealing with proper names, for example, Lila or Elena). Then all uppercase letters are transformed to their lowercase equivalents, and all punctuation marks and digits are removed. This subjective decision could be justified because they can be imposed or modified by the editor or publisher.

## 8.2 Stylistic Mapping of the Contemporary Italian Literature

To produce a stylistic landscape of this large corpus, one can concatenate all novels according to their author's name. After generating 40 large documents, one for each novelist, we can represent them with a reduced set of the topmost frequent word-types (MFWs). According to Biber and Conrad's [31] judgment, a stylistic study should be based on such ubiquitous and frequent forms. As an example, each author will be represented by the relative occurrence frequencies of the 50 MFWs (the precise list of those words appears in Appendix A.5).

Based on these author's profiles, the Principal Component Analysis (PCA) method (see explanations given in Sect. 3.5) projects into two dimensions the relative position of our 40 Italian writers. The resulting graph is reported in Fig. 8.1 corresponding to 85.1% + 4.7% = 89.8% of all the underlying variability. The contribution of the horizontal axis is clearly rather large (85.1%) compared to the second one. The main opposition in the horizontal axis is between styles using more determiners and prepositions on the left (e.g., Ferrante) vs. authors appearing in the right employing the verb *to be*, *to have* and the pronoun *I* (e.g., Morazzoni, Parrella) less than the mean. In the vertical axis, the upper part indicates writers having a preference for the forms *I*, *he's*, *I have*, *me*, for example, Faletti. In the bottom

[4]But both languages belong to the Indo-European family sharing some common characteristics such as the change in form for the first-person pronoun used as a subject (*I* call you) or as an object (You call *me*). For example, one can find *I* and *me* in English, *io* and *mi/me* in Italian, *ich* and *mich/mir* in German, or *je/j* and *me/m* in French.

[5]Available at http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/.

region, one can find novelists (e.g., Piccolo, Carofiglio) using the determiner *the*, and the forms *I'm*, *his/her*, and *of the* more frequently.

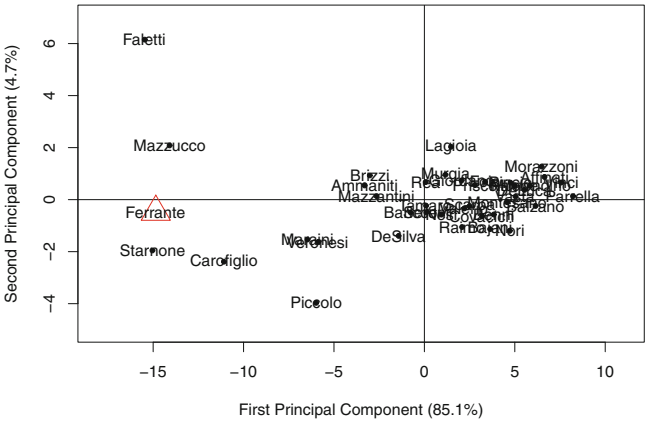


**Fig. 8.1** PCA of Ferrante's profile and all other Italian authors (50 MFWs)

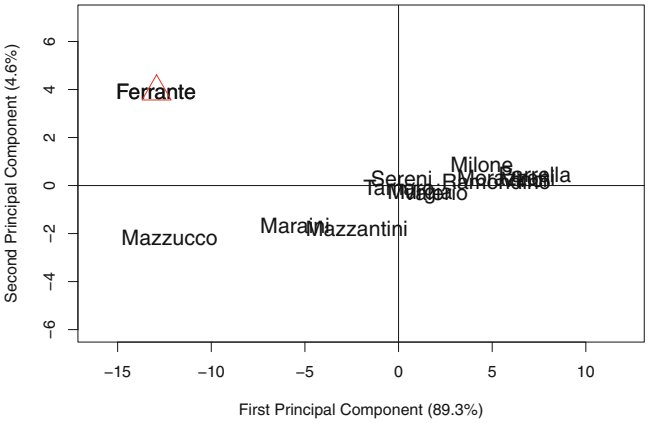


**Fig. 8.2** PCA of Ferrante's profile vs. all other female writers (50 MFWs)

As shown in Fig. 8.1, a large number of Italian authors appear close to the center of the two axes matching the mean style (or more precisely the mean usage of the 50 MFWs). However, a few of them present a distinctive style. In this case, on the left part, one can observe Ferrante (position depicted with a red triangle), Starnone, Mazzucco, or Faletti. For example, Faletti is a well-known novelist of best-sellers, easy to read, but relatively long (e.g., the largest in our corpus (*Io uccido*) was justly written by Falettti). As another example, Carofiglio, who was also a judge,

presents a more complex vocabulary and syntax construction and thus corresponds to a distinctive voice. This first view signals that Ferrante's style is clearly different from the majority of contemporary Italian novelists and shares some similarities with a few names (e.g., Starnone, Carofigio, Mazzucco).

As a second broad view of current Italian literature, Fig. 8.2 depicts the relative positions of all female writers together with E. Ferrante. As for the first PCA, the first axis represents a large proportion of the underlying variance of the 50 MFWs (89.3%). On the left, one can see styles presenting more occurrences of some prepositions together with the definite determiner (e.g., *of the*) and the conjunction *and*. According to [299], the frequent use of determiners and prepositions is one of the characteristics of a male writer. In this view, one can also mention that M. Mazzucco is known as a homosexual woman. This second analysis tends to indicate that Ferrante does not truly have a female style, tending more towards a male one.

In a third PCA representation, only male writers and Ferrante have been chosen. As depicted in Fig. 8.3, this view is strongly related to the first one, with the first axis reflecting 84.6% of the discrepancy between authors. The interpretation of this axis is the same as for Fig. 8.1. Ferrante's style is still positioned close to Starnone, Faletti, and Carofiglio.

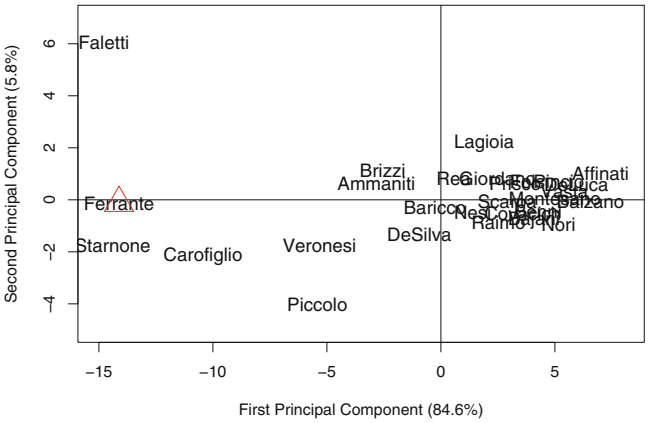


**Fig. 8.3** PCA of Ferrante's profile with all male writers (50 MFWs)

As a last representation, only novelists from Campania (Naples) are depicted in Fig. 8.4. In this picture, male authors are indicated with a blue square, while female ones are depicted with a red triangle. On the right, the writing style employs the auxiliary verb *to be* and *to have* more than the mean together with some prepositions (e.g., *perché* (why), *poi* (then)) and the pronouns *his/her*. On the left, the writing style can be characterized by avoiding using the forms *per* (for), *la*, *il* (the), *non* (not), *alla*, *una*, *un* (a/an).

With the vertical axis, on the top one can find authors using the past tense with *s/he was*, *s/he had*, *his/her*, *she*, or *he* more frequently. On the bottom, the present

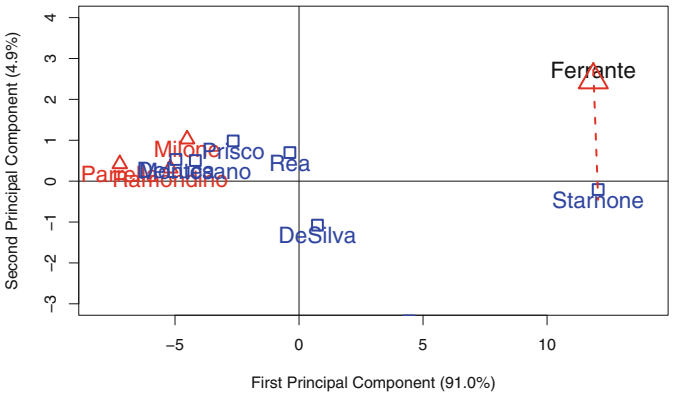


**Fig. 8.4** PCA of Ferrante's profile with all writers from Campania (50 MFWs)

tense is more frequent with *I'm*, *I have*, *s/he has*, and *we*. This axis indicates a stylistic difference between Starnone's style opting more for the present tense (e.g., *sono* (I'm), *è* (s/he is), or *ho* (I have) and *ha* (s/he has)), compared to Ferrante using the past tense more with the forms *s/he had*, *s/he was* (*aveva*, *era*).

As mentioned in Sect. 3.5, one can compute the Euclidian distance between Ferrante and all other authors to discover the author presenting the closest stylistic similarity. In Fig. 8.4, this smallest distance is shown with a dotted red line relating Ferrante to Starnone. The same conclusion can be reached when using the data depicted in Fig. 8.1 with all 40 novelists. In brief, the initial evidence has found that the true author behind Ferrante's novels could be Domenico Starnone, a novelist born in 1943 in Naples. Additional experiments done with this corpus and the PCA approach can be found in [408], [407] leading to the same conclusion.

## 8.3 Delta Model

When applying the Delta model presented in Sect. 3.1, the first step is to define a list of word-types reflecting the differences between writing styles. To achieve this, it is assumed that the frequently occurring terms are used unconsciously by the authors. But the experimenter has some freedom to determine such a list. Usually one can consider the top 50 to 1000 most frequent word-types (MFWs) in the corpus or utilize a predefined list of functional words (e.g., a stop wordlist used in IR systems). We will follow the first strategy and suggest to consider the top 50, 100, 150, 200, 250, 300, 400, and 500 MFWs extracted from our Italian corpus. To define these wordlists, novels for which we want to identify the true possible author have been discarded.

As the Italian language has a richer morphology than English, the lemma can reduce some morphological variations present in the word-tokens (e.g., from the

tokens *amico*, *amica*, *amici*, or *amice*, the same lemma *amico* (friend) is derived). Thus, one can select not the most frequent word-types but the lemmas. We do not expect a large difference between these two solutions because the most frequent terms are determiners, pronouns, prepositions, or conjunctions, word-types that do not vary when considering the lemmas. Some variations can be expected when considering list lengths larger than 200.

In a second step, a profile is generated for all possible novelists (without Ferrante). To achieve this, all books written by each writer are concatenated to build his/her profile (i.e., word-types with their relative frequencies). Of course, not all word-types are taken into account but only the selected ones. Moreover, the value associated with each term is not directly the relative frequency but its standardized frequency or Z score as indicated by Eq. 3.1.

In a third step, a representation of each disputed text is generated according to the same rules. Then one can compute the distance between the doubtful text and all authors' profiles as specified in Eq. 3.2. A ranked list of the different novelists is then presented, from the most likely to the least credible.

Table 8.2 reports the top five names, sorted by the Delta model using the 50 MFWs,[6] with three of Ferrante's novels, namely *L'amore molesto* (her first novel, published in 1992), *L'amica geniale* (the first book of her tetralogy, 2011), and *Storia della bambina perduta* (the last book of her tetralogy, 2014).

**Table 8.2** Ranked lists produced by the Delta model (50 MFWs, profile-based approach)

| Rank | Distance | Author | Distance | Author | Distance | Author |
|---|---|---|---|---|---|---|
| | *L'amore molesto* | | *L'amica geniale* | | *Storia bambina perduta* | |
| 1 | 0.495 | Starnone | 0.522 | Starnone | 0.572 | Starnone |
| 2 | 0.657 | Brizzi | 0.582 | Balzano | 0.671 | Balzano |
| 3 | 0.666 | Milone | 0.650 | Milone | 0.691 | Veronesi |
| 4 | 0.695 | Sereni | 0.705 | Veronesi | 0.703 | Carofiglio |
| 5 | 0.724 | Tamaro | 0.711 | Prisco | 0.708 | Milone |

In Table 8.2, the author with the closest writing style to Ferrante is always Starnone. This is the case not only for the three novels depicted in Table 8.2, but for all seven of Ferrante's books. Appearing in lower ranks, other names appear that could also be possible author behind Ferrante such as Milone (a female writer from Naples), Prisco (a male writer also from Campania), or Veronesi. The name Carofiglio appears from time to time because he shares some stylistic similarities with Ferrante as shown in Fig. 8.1 or 8.3.

Applying the same model with the 100, 150, 200, 250, 300, 400, or 500 most frequent word-types (MFWs) or lemmas (MFLs), all seven of Ferrante's novels are

[6]Even if Ferrante's novels are ignored to generate the top 50 MFWs, these 50 words are the same as those appearing in the wordlist used in the previous section (of course without appearing in the same order).

always assigned to Domenico Starnone. A relatively strong evidence in favor of this name.

However, one cannot be 100% sure that the true author behind Ferrante's novels is really D. Starnone. One must conduct additional experiments with the Delta method to verify and corroborate this conclusion. To achieve this, Table 8.3 depicts the ranked lists of the most probable novelists for three of Ferrante's novels. These lists were generated with the 50 MFWs built by ignoring both Ferrante's and Starnone's novels. As possible author, the name Starnone was also removed from the candidates. The idea is to verify whether another name would appear in a systematic way when ignoring Starnone.

The resulting lists are strongly correlated to those reported in Table 8.2, except for the first name (Starnone is absent in Table 8.3). Clearly, for the seven Ferrante books, several distinct names appear in the first position (twice Balzano, twice Brizzi, twice Milone, and once Veronesi). Thus, one cannot see a systematic result leading to a possible other name.

**Table 8.3** Ranked lists produced by the Delta model but without Starnone (50 MFWs, profile-based approach)

| Rank | Distance | Author | Distance | Author | Distance | Author |
|---|---|---|---|---|---|---|
| | *L'amore molesto* | | *L'amica geniale* | | *Storia bambina perduta* | |
| 1 | 0.650 | Brizzi | 0.575 | Balzano | 0.663 | Balzano |
| 2 | 0.658 | Milone | 0.642 | Milone | 0.687 | Veronesi |
| 3 | 0.687 | Sereni | 0.700 | Veronesi | 0.696 | Carofiglio |
| 4 | 0.719 | Carofigio | 0.703 | Prisco | 0.701 | Milone |
| 5 | 0.720 | Tamaro | 0.712 | Sereni | 0.753 | Prisco |

As another verification, one can analyze the distance values of the first three ranks in Tables 8.2 and 8.3. As a result, one can discover interesting evidence about the strong relationship between Ferrante's and Starnone's writing styles. When computing the differences between the first and the second distance, one can observe that those differences are larger in Table 8.2 than in Table 8.3. For example, in Table 8.2 (*L'amore molesto*), the absolute difference between the first two ranks is $|0.495 - 0.657| = 0.162$ (or 32.7% of the smallest distance, i.e., 0.495). The divergence between the second and the third is $|0.657 - 0.666| = 0.009$ (or 1.4%). This comparison indicates that the first answer is clearly more probable than the other writers occurring in the ranked list. In Table 8.3, and with *L'amore molesto*, one can only observe a small difference between the first two authors, namely $|0.650 - 0.658| = 0.008$ (or 1.2%). In this context, the Delta model is not able to clearly discriminate one author over the others.

As a third complementary analysis, we suggest to detect who presents the closest stylistic similarity with Starnone's novels. In other words, we reverse the attribution procedure. Stranone's books are not used to define the 50 MFWs but Ferrante is considered as one possible novelist. From the ten books authored by Starnone,

the ranked list for three of them are reported in Table 8.4. We repeated the same computation for 100, 150, 200, 250, 300, 400, and 500 MFWs.

**Table 8.4** Ranked lists produced by the Delta model with three of Starnone's novels (50 MFWs, profile-based approach)

| Rank | Distance | Author | Distance | Author | Distance | Author |
|---|---|---|---|---|---|---|
| | *Ex Cattedra* (1987) | | *Via Gemito* (2000) | | *Lacci* (2014) | |
| 1 | 0.724 | Covacich | 0.541 | Ferrante | 0.639 | Ferrante |
| 2 | 0.772 | Maraini | 0.614 | Brizzi | 0.656 | Raimo |
| 3 | 0.828 | Balzano | 0.616 | Milone | 0.692 | Carofiglio |
| 4 | 0.828 | Benni | 0.643 | Nesi | 0.703 | Balzano |
| 5 | 0.855 | Brizzi | 0.652 | Giordano | 0.733 | Veronesi |

From the full result of these 10 books × 8 feature sets = 80 experiments, the name Ferrante occurs in the first rank 60 times, Veronesi 9 times, Maraini 7 times, Carofiglio twice, and both Covacich and Brizzi once. Ferrante is not always the closest to Starnone's style and not always in the second or third position as shown in the first column of Table 8.4.

At this point, a pertinent source of stylometric variation must be put forward: the publication date. Stranone's novel *Ex cattedra* appears in 1987 and two other novels (in 1989 and 1991) have been published before the first book authored by Ferrante (1992). The style of an author is not fully stable over his life [121] and change does occur. For example, in Table 8.4, one can verify that the names appearing from the second to the fifth position are different when considering three novels published within a time gap of around 15 years. Moreover, the phenomenon related to an increased speed of style variation during the twentieth and twenty-first century was already mentioned in [328].

To partially ignore this publication time difference, one can ignore the three Starnone novels published before 1992. With the seven remaining books, the stylistic similarity between Starnone and Ferrante is more evident. Considering 50, 100, 150, 200, 250, 300, 400, and 500 MFWs, one can find the name Ferrante is the first rank 54 times (over 7 books × 8 experiments = 56). In the remaining two other cases (with 50 MFWs), one can find Carofiglio, a name already found similar to Starnone in Figs. 8.1 and 8.3.

In conclusion, the Delta model with its additional experiments indicates that the most probable novelist of Ferrante's novels is Domenico Starnone. This finding assumes that the true author is present in our corpus composed of 39 possible Italian writers (closed-set assumption). Moreover, based on the differences in the distance values between the first two ranks, one can have reasonable confidence that this conclusion is also true when assuming an open-set assumption. In addition, looking at the closest style to Stranone's books, one can usually find Ferrante's name, particularly when considering the latest novels published by Starnone. Of course, one can propose other tests based on the Delta model, for example, by replacing

the author's profiles by novel profiles, see [347] or by using the Rolling Delta (see Sect. 7.5) [101].

## 8.4 Labbé's Intertextual Distance

With Labbé's intertextual distance [225] exposed in Sect. 3.3, the stylistic features are extracted from the entire vocabulary. To be precise, the word-types appearing just once or twice have been ignored in this representation. This pruning approach tends to reduce the number of terms to be selected by 50% (see Zipf's law in Sect. 2.3). Moreover, to avoid considering several related terms, Labbé suggests to represent texts with their lemmas instead of tokens. As the Italian morphology is more complex than the English one, this aspect will regroup variations in gender (masculine or feminine), in number (singular or plural), in tense and person (for verbs) into the same entry. In Italian, such variations occur for both nouns, adjectives, and, as for the English language, for verbs. Considering the entire corpus, the vocabulary size is 83,326 lemmas. When taking into account only lemmas depicting a frequency larger than two, the vocabulary size is reduced to 40,648 (a decrease of 51.2%). Moreover, the text length used in this case is larger than 10,000 words, which strengthens the conclusion.

In our experiments the punctuation symbols are included. The most frequent one is the comma followed by the full stop, a frequency that is strongly related with the mean sentence length. Of course, this choice is subjective and another analysis could ignore all punctuation marks or consider only some of them. With these considerations, the stylistic representation used in this section is rather different from the Delta model presented previously.

As another difference with the two previous methods, Labbé's intertextual distance is computed for all pairs of novels and ranked from the smallest to the largest distance. Thus, this approach does not employ an author's profile, but only novel representations.

Table 8.5 gives an excerpt of such ranked lists reflecting the distance with two of Ferrante's novels, on the left, *L'amica geniale* (2011), and on the right *Storia di chi fugge e di chi resta* (2013). Clearly, the closest books to these two novels have been written by Starnone. For the first one, four books authored by Starnone appear in the first four ranks, for a total of seven in the top ten positions. For the second case, seven novels written by Starnone occur in the first ten ranks, with six appearing in the first six ranks. For the other Ferrante books, one novel authored by Starnone always appears in the first rank.

Therefore, using a large number of lemmas (40,648 lemmas) to reflect each author's style, the same conclusion is reached: Domenico Starnone is the true writer of Ferrante's works.

When inspecting the minimal distances, in some cases the stylistic distance is lower than 0.2 (e.g., the second novel written by Ferrante in Table 8.5 depicted three examples). When considering texts of high quality with lengths larger than

**Table 8.5** Ranked list of Labbé's distances for two of Ferrante's novels (lemmas)

| Rank | Dist. | *L'amica geniale* | | Dist. | *Storia di chi fugge e di chi resta* | |
|---|---|---|---|---|---|---|
| 1 | 0.201 | Starnone | *Via Gemito* | 0.184 | Starnone | *Lacci* |
| 2 | 0.206 | Starnone | *Lacci* | 0.194 | Starnone | *Prima esecuzione* |
| 3 | 0.208 | Starnone | *Prima esecuzione* | 0.197 | Starnone | *Autobiografia erotica* |
| 4 | 0.213 | Starnone | *Autobiografia erotica* | 0.208 | Starnone | *Via Gemito* |
| 5 | 0.215 | Sereni | *Una storia chiusa* | 0.211 | Starnone | *Scherzetto* |
| 6 | 0.221 | Starnone | *Il salto con le aste* | 0.215 | Starnone | *Il salto con le aste* |
| 7 | 0.223 | Milone | *Il silenzio del lottatore* | 0.219 | Veronesi | *Caos calmo* |
| 8 | 0.225 | Starnone | *Fuori registro* | 0.222 | Sereni | *Una storia chiusa* |
| 9 | 0.231 | Veronisi | *Caos calmo* | 0.225 | Veronesi | *Terre rare* |
| 10 | 0.233 | Starnone | *Eccesso di zelo* | 0.231 | Starnone | *Fuori registro* |

10,000 words, this is a strong indication that the same author wrote the two novels. This clearly appears with *Storia di chi fugge e di chi resta* and three of Starnone's books (*Lacci* (2014), *Prima esecuzione* (2007), and *Autobiografia erotica di Aristide Gambìa* (2011)). The novel *Lacci* also presents a distance value smaller than 0.2 with the second and the last book of the *My Brilliant Friend*'s saga. In addition, Starnone's novel *Prima esecuzione* also presents a distance value smaller than 0.2 with the second and third of Ferrante's books (*I giorni dell'abbandono* (2002) and *La figlia oscura* (2006)). No other novelist depicts such a strong and close intertextual distance.

As a way to confirm this strong relationship between Starnone and Ferrante, one can reverse the attribution procedure. What are the books closest to Starnone's novels? A partial answer is reported in Table 8.6 with two of Starnone's books.

**Table 8.6** Ranked list of Labbé's distances for two of Starnone's novels (lemmas)

| Rank | Dist. | *Prima esecuzione* (2007) | | Dist. | *Lacci* (2014) | |
|---|---|---|---|---|---|---|
| 1 | 0.185 | Ferrante | *I giorni dell'abbandono* | 0.181 | Ferrante | *Storia della bambina* |
| 2 | 0.194 | Ferrante | *La figlia oscura* | 0.184 | Ferrante | *Storia di chi fugge e di ...* |
| 3 | 0.194 | Ferrante | *Storia di chi fugge e di chi* | 0.198 | Ferrante | *Storia del nuovo ...* |
| 4 | 0.207 | Ferrante | *Storia del nuovo cognome* | 0.206 | Ferrante | *L'amica geniale* |
| 5 | 0.208 | Ferrante | *L'amica geniale* | 0.222 | Veronesi | *Caos calmo* |
| 6 | 0.208 | Ferrante | *Storia della bambina* | 0.223 | Veronesi | *Terre rare* |
| 7 | 0.214 | Raimo | *Latte* | 0.224 | Ferrante | *I giorni dell'abbandono* |
| 8 | 0.229 | Milone | *Il silenzio del lottatore* | 0.229 | Scarpa | *Le cose fondamentali* |
| 9 | 0.231 | Sereni | *Una storia chiusa* | 0.229 | Ferrante | *La figlia oscura* |
| 10 | 0.235 | Tamaro | *Va' dove ti porta il cuore* | 0.231 | Tamaro | *Va' dove ti porta il cuore* |

In this experiment, the first three positions are always occupied by a novel authored by Ferrante with the exception of the last Starnone book (*Scherzetto* (2016)) for which the second place presents a book written by Raimo (*Il peso della*

*grazia* (2012)). This is more evidence about the strong stylistic similarity between Ferrante's and Starnone's writings.

Can we infer more from the intertextual distance between the two novels? And especially when such a distance is rather small, for example, 0.201 depicted in Table 8.5 with the book *L'amica geniale* (Ferrante, 2011) and *Via Gemito* (Stanone, 2001) or in the same table with *Lacci* (Starnone, 2014) appearing twice with a small distance value to Ferrante's novels?

When considering Labbé's intertextual distance between each pair of novels, one can consider two distributions of values, those computed when faced with two books written by the same author or when considering two distinct novelists. As the novels written by Ferrante could have been authored by one of the remaining 39 writers, we need to exclude Ferrante's books for this computation, leading to $150 - 7 = 143$ texts.

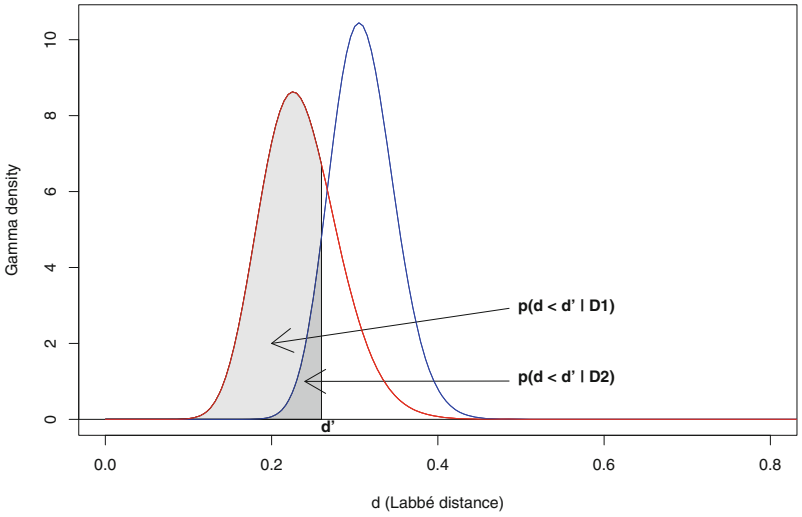


**Fig. 8.5** Labbé's intertextual distance viewed as two Gamma distributions (lemma-based text representation)

According to the number of authors and novels in the corpus (see Table 8.1), one can count 258 distances derived from pairs of novels authored by the same person and 9895 from pairs of texts written by two distinct authors. Instead of displaying the corresponding histograms of both distributions, one can model each of them with a Gamma distribution[7] [342] as depicted in Fig. 8.5. In this diagram, the distribution denoted D1 corresponds to pairs of books written by the same person. Clearly

[7]Using the histograms directly, the smallest distance between two novels written by two distinct authors is 0.210. This implies that values smaller than 0.210 between two authors are impossible. A rather extreme position (see a related discussion about smoothing procedures in Sect. 3.2).

the resulting distance values are smaller compared to distribution D2 (distances computed with two novels authored by two different writers).

With the smallest distance values, the distribution D1 is more appropriate or, in other words, those values have a higher probability to come from D1. However, large distance values agree more closely to distribution D2.

Thus, given an intertextual distance value, one can estimate the probability that this value (e.g., 0.201 in our example) could be observed between two texts written by the same person according to distribution D1. In addition, one can compute the probability that this distance is related to distribution D2 (two distinct authors) (the details are provided in [342]).

To estimate the probability that a distance value $d = 0.201$ links two texts written by the same author, $p(d < 0.201|D1)$ and $p(d < 0.201|D2)$ are computed. The sum of these two probabilities is used as a normalization constant. With this method, the probability that $d = 0.201$ links two texts written by the same person is given by $p(d < 0.201|D1)$ and is equal to 99.7% compared to the alternative solution p(d $< 0.201|$ D2) = 0.3%.

Thus, as shown previously in Table 8.6, the first positions present a book authored by Ferrante. Moreover, the computed distance is sometimes smaller than 0.2, indicating that the same writer is behind the two books.

## 8.5 Zeta Test

The Zeta test (see Sect. 7.1) suggested by Burrows [47] can be applied to analyze the lexical proximity between Ferrante's and Starnone's novels (see also [65, 320]). This approach focuses on the terms appearing recurrently in the text passages written by the first author (denoted A forming the base set) and rarely in excerpts written by another writer (denoted B and forming the counter set). The feature selection is based on the presence and absence of terms (e.g., isolated words in our experiment), not on their occurrence frequencies. In addition, instead of considering entire novels, each book is decomposed into non-overlapping chunks of size $k$ (e.g., $k = 4000$ in the current study). Such a reduced size is more appropriate when considering only the presence and absence of terms and can generate several instances from a single book.

When a term occurs in all chunks written by A and never in B, its Zeta score reaches the maximum value of 2. However, when the word occurs in all passages written by B and never with A, its discriminative value is 0. Finally, when a word is very frequent and appears in all chunks, its Zeta weight is 1.

After computing the Zeta score for all words and sorting them, the terms associated with author A appear on the top, with a value larger than 1.0. On the bottom part, one can identify words ignored (or used rarely) by A and occurring frequently with the second novelist B (with a value smaller than 1.0). In this study, the top 200 words having a Zeta weight larger than 1.0 have been selected to form the vocabulary specific to A and the lower 200 words (with a weight smaller than

1.0) to determine the words associated with B. Examples of such words are reported in Table 8.7 with Starnone forming the base set and Brizzi the counter set.

**Table 8.7** Words occurring regularly in Starnone's or Brizzi's text passages

| | Starnone | | Brizzi | |
|---|---|---|---|---|
| Rank | Zeta | Terms | Zeta | Terms |
| 1 | 1.66 | *perciò* (therefore) | 0.12 | *fra* (between) |
| 2 | 1.66 | *tra* (between) | 0.33 | *domandò* (he asked) |
| 3 | 1.57 | *accanto* (next) | 0.39 | *domandai* (I asked) |
| 4 | 1.53 | *spesso* (often) | 0.39 | *mica* (surprise[a]) |
| 5 | 1.48 | *qua* (here) | 0.44 | *neppure* (neither) |
| 6 | 1.47 | *frasi* (phrases) | 0.49 | *paio* (pair) |
| 7 | 1.47 | *esempio* (example) | 0.50 | *maniera* (way) |
| 8 | 1.44 | *chiese* (asked/churches) | 0.51 | *max* (a name) |
| 9 | 1.44 | *soprattutto* (mostly) | 0.52 | *considerò* (considered) |
| 10 | 1.42 | *sicché* (so) | 0.52 | *spiegò* (he explained) |

[a]The word *mica* is used to denote surprise, usually in a dialogue

Finally, based on these two wordlists, one can visualize the lexical proximity between a given novel (composed of a set of passages) and both authors A (or Starnone in our example) and B (or Brizzi). To achieve this, the target text is divided into non-overlapping chunks (of size $k$). For each passage, the percentage of words appearing in both lists indicates the two coordinates. Similarly, chunks of texts written by A and B can be decomposed and each passage can be added into the graph.

In Fig. 8.6, the 200 words more specific to Starnone are used to define the X-coordinate, while the terms appearing more frequently in novels written by Brizzi define the Y-coordinate. Starnone's cloud is composed of 192 chunks ("x" depicted in red in Fig. 8.6), while Brizzi's one contains 105 points shown as blue triangles in Fig. 8.6. In addition, the gravity center of both clouds appears as a solid colored circle with the letter "S" or "B."

The tested novel is *L'amica geniale* (Ferrante) subdivided into 30 passages (of 4000 words). As depicted in Fig. 8.6, all these points (shown as green dots) appear in or very close to Starnone's cloud. The gravity center for this novel is depicted by a solid colored circle with the letter "F" located closely to Starnone's center.

Visually, this graph indicates a strong lexical proximity between novels written by Starnone and the novel *L'amica geniale* (Ferrante, 2012). The same book appears distant from Brizzi's cloud even if this author has some stylistic relationship with Ferrante (see Tables 8.2, 8.3, and 8.5). Repeating the same experiment with other Ferrante's novels or other writers than Brizzi, the same conclusion can be reached: Starnone is the closest author to Ferrante's novels.

As a second experiment, the same novel (*L'amica geniale*, Ferrante) was reused but this time with books written by Milone (35 points) and Veronesi (118 chunks

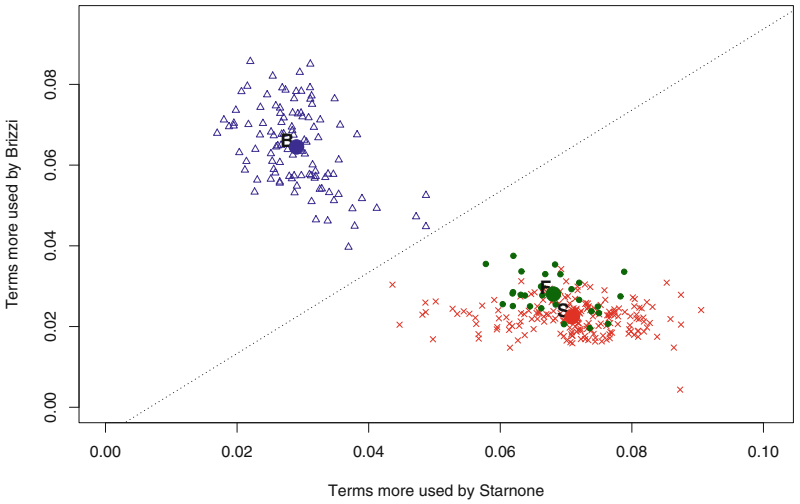


**Fig. 8.6** Zeta representation with all Starnone vs. Brizzi's novels together with Ferrante's with *L'amica geniale*

of 4000 words). Both authors have been found to share some similarities with Ferrante's style as indicated by Tables 8.2, 8.3, and 8.5.

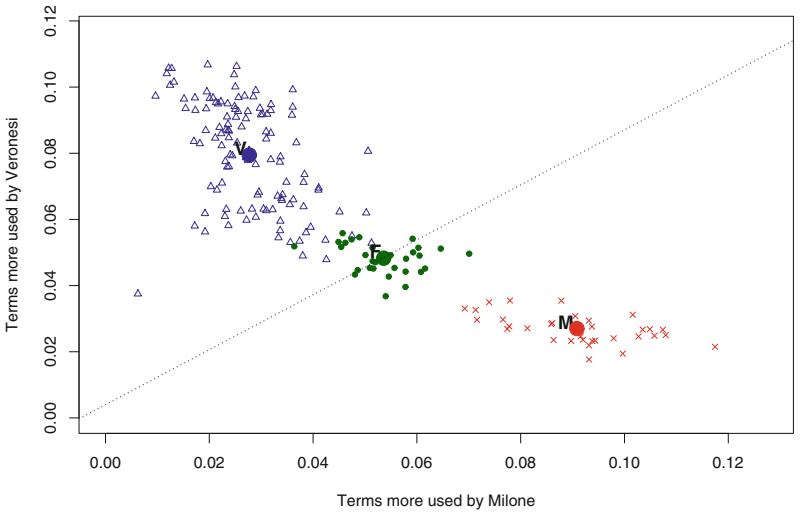


**Fig. 8.7** Zeta representation with Milone and Veronesi's novels together with *L'amica geniale*

As depicted in Fig. 8.7, one can observe three distinct clouds of points. On the top left, the points corresponding to Veronesi's novels, on the bottom right, the

passages written by Milone, and in the middle, the cloud defined by the novel *L'amica geniale*. The three gravity centers occupy clearly distinct positions. This Zeta visualization indicates stylistic differences with both Milone and Veronesi when compared to Ferrante's novel *L'amica geniale*.

## 8.6 Qualitative Analysis

When applying our different attribution models, we implicitly admit that the lexical choice and the term frequencies can reveal each author's distinctive style. More explicit reasons justifying the strong lexical similarity between Starnone and Ferrante can be found when inspecting the word usage of these two authors, as compared to the others. Focusing on frequent words, one can assume that those terms are employed with similar frequencies by all writers. Then, their occurrence frequencies must be similar to the proportion of the novels written by each author. For example, Starnone's books represent 6.4% of the corpus and Ferrante's 6.5%. When a word frequency approximatively reaches the 6.4% (Starnone) or 6.5% (Ferrante), it appears with the expected frequency for this author. However, when the frequency is clearly higher (or lower), the writer tends to use it more (or less) often than the other novelists.

Our first example is the word-type *padre* (father), occurring 9815 times (100%) in the corpus. Compared to all the other novelists, this word-type is proportionally more frequent in Ferrante's novels (8.5% for 833 occurrences) and in Starnone's writings (11.9% for 1170 occurrences). A similar distribution can be observed for the word-type *madre* (mother): its frequency in the corpus is 8246, with 1104 in Ferrante's works (13.4%) and 762 in Starnone's (9.2%).

Additional examples can be found and Table 8.8 depicts other word-types such as *perciò* (therefore) occurring 1263 times in the entire corpus, with 222 occurrences (17.6%) in Ferrante's novels and 254 (20.1%) in Starnone's. In the last column of Table 8.8, the chi-square test has been applied, to verify whether the word-type distribution differs significantly between the authors (all $p$-values $< 0.1\%$) [285]. One can also see that the terms *perciò* and *frase* were also detected by the Zeta method as shown in Table 8.7.

As a unique case, the word-type *persino* (even) can also be spelled as *perfino*. For both Ferrante and Starnone, the preferred spelling is *persino* (used 266 vs. 20 times for *perfino* in Ferrante's writings, 205 vs. 18 times in Starnone's novels). This pattern can also be found in the works of a few other writers, such as Prisco (132 occurrences of *persino*, 1 of *perfino*). Some novelists employ only one form (e.g., Baricco with *perfino*, Tamaro with *persino*), while others omit both words (e.g., Covacich, Parrella) or use them only rarely (e.g., De Luca or Balzano, with a single occurrence of *perfino*). However, this word-type is clearly overused by both Ferrante and Starnone.

As another lexical analysis, one can detect words that are only employed by these two writers, such as *contraddittoriamente* (15 occurrences, contradictory),

**Table 8.8** Examples of words occurring more frequently in Ferrante's and Starnone's novels

| Word | Corpus | Ferrante (6.5%) | Starnone (6.4%) | Significant? |
|---|---|---|---|---|
| *padre* (father) | 9815 | 833 (8.5%) | 1170 (11.9%) | yes |
| *madre* (mother) | 8246 | 1104 (13.4%) | 762 (9.2%) | yes |
| *perciò* (therefore) | 1263 | 222 (17.6%) | 254 (20.1%) | yes |
| *persino* (even) | 1351 | 266 (19.7%) | 205 (15.2%) | yes |
| *temere* (fear) | 1345 | 274 (20.4%) | 207 (15.4%) | yes |
| *tono* (tone) | 2135 | 421 (19.7%) | 286 (13.4%) | yes |
| *gridare* (shout) | 2201 | 399 (18.1%) | 303 (13.8%) | yes |
| *monstrare* (to show) | 2271 | 384 (16.9%) | 310 (13.7%) | yes |
| *content* (happy) | 1665 | 280 (16.8%) | 227 (13.6%) | yes |
| *brutto* (ugly) | 1893 | 327 (17.3%) | 243 (12.8%) | yes |
| *frase* (phrase) | 2182 | 334 (15.3%) | 312 (14.3%) | yes |

*giravite* (13 occurrences, screwdriver (more often named *cacciavite*)), *studenti* (10 occurrences, students), and *soffertamente* (8 occurrences, by suffering). An interesting example is the word-type *malodore* (17 occurrences, stink), appearing with this spelling in both Ferrante's and Stanone's novels; however, the same meaning can appear as *mal odore* or *maleodore*. These last two spellings appear in other novels, but never under Ferrante's or Starnone's authorship.

As a third stratum of word frequency, one can consider word-types with low occurrence frequency in the whole corpus, specifically those occurring clearly more often in works by Starnone and Ferrante than in those by other Italian authors. For example, the term *minutamente* (minutely) occurs 28 times in Ferrante's novels, 14 times in Starnone's writings, and 3 times in the rest. With *tassare* (to tax), one can observe something similar: 22 for Ferrante, 10 for Starnone, and 3 times for the others. The word-type *reattività* (reactivity) occurs 22 times in the whole corpus, with Ferrante employing it 6 times and Starnone 13 times. Our last example relates to dialect usage, with the word *strunz* (shit). This term does not belong to the classical Italian language (in which it is spelled as *stronzo*), but corresponds to a Neapolitan dialect form. The distribution of occurrence for this word is as follows: 18 in Ferrante's novels, 63 times in Starnone's writings, and 4 times for all the others (twice in De Silva's novels and twice in Raimo's novels).

## 8.7 Conclusion

Our finding is based only on the writing and it does not imply that the whole story was produced by a single person. One can imagine a collaboration between two (or more) persons for elaborating the scenario, generating some characters, and creating some dialogues or sentiments.

As discussed in this chapter, the real author behind Ferrante's books is certainly Domenico Starnone. One can argue that A. Raja, a translator and Starnone's wife, was not included in the author list. As shown in [330], one can detect some of her stylistic idiosyncrasies and compared them to Ferrante's. More generally, under certain conditions, it is possible to detect the translator's own signal in a work [332]. In this case too, the analysis confirms Starnone's writing style behind Ferrante's novels. This conclusion is also reached by other authorship methods such as compression-based [233], by profiling strategies [265] or by visualizing author clusters [101].

According to past evaluations, one can suppose that each previously described model has an accuracy rate of 0.8 (or 80%), a rather conservative value. Consequently, the chance of providing an incorrect assignment is 0.2 (or 20%). Assuming that their results are independent, the chance that two attributions are incorrect is $(1-0.8)\times(1-0.8)=(1-0.8)^2=0.04$ (or 4%). With four models (PCA, Delta, Labbé, Zeta), this probability decreases to $(1-0.8)^4=0.0016$ or 1.6 in 1000. The chance that such a systematic error might occur is very low (but not impossible).

Finally, in this context many questions are still without a clear answer. For example, how can we explain Ferrante's worldwide success? Is it due to the adaptation of her first two books into movies?[8] Or with the *My Brilliant Friend*'s tetralogy, a story about two girls in Naples in the fifties and the social dynamics of Italy after the Second World War? One can also assume that the mystery about her real identity might explain a larger press coverage and thus a larger popularity or simply because she has a real distinctive style (as shown in Fig. 8.1). And in this latter case, can we design a computer program to identify best-sellers or a successful style? A related question is to understand how an author proceeds to differentiate (or tries to do so) his own style with the style he wants to associate with an *alter ego* [101].

[8] *L'Amore molesto* (1992) and *I Giorni dell'Abbandono* (2002) appear under the same name as movies directed, respectively, by M. Martone and R. Faenza.

# Chapter 9
# Author Profiling of Tweets


The social networks, and more precisely Twitter, form the background of our second case study. During the first decade of the twenty-first century, various applications and social platforms have been launched (e.g., blogging,[1] MySpace in 2003, Facebook in 2004, Twitter in 2006, Instagram 2010, etc.) transforming the habits of Internet users by allowing them to be information producers. While instant messaging systems were mainly dedicated to send short texts between two persons (e.g., emails), social media allows the writer to have a larger audience (e.g., chats, blogs, tweets). And not only individuals but also companies and various organizations (e.g., political parties, NGO, ...) consider social networks as a main stream to communicate with their customers, employees, stakeholders, and the public in general.

When studying the language used in such social media, linguistics found that this new communication channel is distinct from the traditional written or oral form [74, 77, 256]. Within social networks, the spelling, orthography, and grammar are less respected than in a formal document [77]. Moreover, if the Internet is characterized by its hyperlinks (e.g., http://www.nytimes.com), the social networks allow the author to include them to establish semantical-based relationships between messages or posts. In addition, through the support of hashtags (e.g., #love, #photooftheday), the sender could specify the main topics of his post. In Twitter, the user could also utilize mentions (e.g., @rogerfederer) to draw attention to another Twitter account. As a new additional feature, the user can retweet a previous one they found interesting. In such cases, the letters "RT" appear in the front of a re-posted message (Twitter will automatically add the symbol 🔁). Finally, with the available emojis, the writer could indicate some of his or her state of mind or emotions to the reader (e.g., 😳, 😜). As the number of emojis increases each year,

[1] Even if the first blog system was Link.net created by J. Hall in 1994, the platform Blogger (E. Williams and M. Hourihan, 1999) really popularized this practice.



J. Savoy, *Machine Learning Methods for Stylometry*,
https://doi.org/10.1007/978-3-030-53360-1_9

they can also be used to shorten the message by indicating places, activities, foods, animals, or time (e.g.,  ,  ,  ,  ,  ,  ,  ).[2]

With these new possibilities, it is not clear how the different categories of writers will employ them and with which intensity. Therefore, the CLEF PAN 2019 evaluation campaign organizers have generated a tweet corpus written in English and Spanish to investigate those questions. However, tweet metadata (account age, number of likes, followers or retweets, number of tweets per day, etc.) is not available. In previous studies, such information has been found pertinent to discriminate between tweets sent by bots or humans [56].

The rest of this chapter is organized as follows. Section 9.1 provides a description of this corpus. Section 9.2 focuses on the identification of two types of senders, namely bots or humans. Our purpose is not to propose a new and more effective solution to this task. In this perspective, many distinct strategies have been suggested and evaluated [314] (see also [321]). Our main concern is to show how stylometric methods presented in the previous chapters could be applied to solve this question and to provide useful explanations about the stylistic differences between the two categories. In a similar way, Sect. 9.3 describes methods able to enlighten stylistic variations according to the author's gender (man vs. woman). Moreover, this part illustrates how the specific features available with Twitter are used differently by men and women. Finally, a conclusion draws the main findings of this case study.

## 9.1 Corpus and Research Questions

During the CLEF PAN 2019 evaluation campaign, two main research topics were submitted to the participants. First, can we automatically detect whether a set of tweets have been sent by a bot or a human? Second, when this set of tweets has been written by a human, can the computer identify the author's gender?

The first problem is related to the ubiquitous presence of spam posts and, in particular, irrelevant tweets. Such messages are sent to influence users for commercial, political, or ideological purposes, as well as to propagate fake information or news. During an election or a marketing campaign, soft bots could be programmed to write and spread posts to favor or undermine the image of a product or the reputation of a candidate [49, 116, 173, 223, 380, 414]. Recently, Twitter introduced a fact-checker mechanism that adds a short label ("Get the facts about ...") when identifying doubtful information present in a tweet. From this tag, the users can be redirected to articles provided by CNN, *The Washington Post*, and *The Hill* (newspaper), as well as with a page summarizing the findings of fact-checkers [96]. Misinformation is one example of spam tweets, but unsolicited ads or irrelevant contents are other examples. One can suspect that the writing style of such machine-generated tweets

[2] As each emoji is a small picture, they could appear differently according to the device (smart phone, computer, printed book).

could significantly differ from tweets written by humans [115, 145]. If it is the case, what are the stylistic features that can distinguish between the two categories and can this task be performed automatically?

The discrimination between bots and humans could be related to spam filters present in all mailboxes. The underlying classifier must discard irrelevant emails, as well as spam ones. But in this case, the target email could be written by a human or generated by a computer. Moreover, according to the content, some emails could be junk for one person and legitimate for another. In our context, the identification of bot versus human is strongly related to the stylistic aspects, not directly to the content itself.

The second research question is more usual as it was recognized that men and women tend to present distinct writing styles [299] and even on social media [297, 353], or more specifically on Facebook [294].

The corpus created by the CLEF PAN organizers is composed of tweets generated by bots, men, and women. The identification of bots was performed according to Twitter accounts identified as such by previous studies (e.g., [69, 413]) and complemented by dedicated searches (using the query "I'm a bot"). The human accounts correspond mainly of sources used during previous CLEF PAN evaluation campaigns [315]. In addition, some manual verifications were performed to ensure the overall quality of the data and the identification of the right label. Some examples of tweets are reported in Table 9.1. Of course, when performing the first task, the human category regroups tweets written by a man or a woman.

**Table 9.1** Examples of tweets sent by bots, men, or women

| Tweet | Category |
|---|---|
| sharing #suaju Could Surgeons Operate Using the Internet? - Voice of America #internet #web https://www.t.co/cdzSts6mwc | Bot |
| RT @Talnts: Are you a model, an Athlete, a Photographer or a Musician? Well, Talnts is the right network for you!... https://www.t.co/d2J2kGgdzd | Bot |
| 2 Years, 6 Months, 16 Days, 19 Hours, 24 Minutes, and 22 Seconds | Bot |
| Japan stuns Colombia in World Cup opener https://www.t.co/iFOUvalUZb https://www.t.co/SdRtuvjL9a | Bot |
| @dawnsgeddes Loved it too! Pretty dark and difficult but SO well put together and beaut | Woman |
| Les tartes! @pieholevan so delicious, thank you 😍 https://www.t.co/FEr07YPXwJ | Woman |
| RT @tash_shaya: Bruh. How do we, the people that live here, not know that there is a virus in our country? 😂 https://www.t.co/7fhmZGqg8L | Man |
| State won, lions won not a bad football weekend | Man |

As the task could be really difficult when considering a single tweet (limited to 140 characters in our corpus), the classification decision must be taken based on a set of 100 tweets sent by the same source. We will call such a set of 100 tweets a document or an instance.

This CLEF PAN corpus can be subdivided into two parts. The training set to allow the methods to learn the distinctive stylistic characteristics for both categories, and the test sample reserved to evaluate the effectiveness of the proposed classification approach. Exactly the same subdivision was present during the CLEF PAN 2019 campaign.

Table 9.2 reports the number of documents in both samples according to the different categories. Clearly, this corpus is balanced when considering the first task (bots vs. humans) or the second (men vs. women). This corpus is freely available (see `www.pan.webis.de`).

**Table 9.2** Statistics about the tweet corpus (in number of sets of 100 tweets)

| | Bots | Humans | Male | Female |
|---|---|---|---|---|
| Training | 2060 | 2060 | 1030 | 1030 |
| Test | 1320 | 1320 | 660 | 660 |
| Total | 3380 | 3380 | 1690 | 1690 |

The distinction between bots or humans seems relatively clear, but the limitation to two genders could raise some concerns. When taking account of more categories (i.e., LGBT), the sample generation is rather more complex and certainly the number of available observations would be smaller.

When considering the bots, one must not think about a homogeneous class because one can see four different types, namely feed, template, quote, and advanced bots. The first type corresponds to an automatic Twitter account sending retweets of news on specific or predefined topics (e.g., to support Trump's policies). With template bots, the system knows a given pattern (a template) that can be filled with certain information (such as job offers shown in Table 6.6). Quote bots reproduce passages or quotes from famous books (e.g., the Bible), songs, or sentences spoken by celebrities, etc. Advanced bots have been designed based on more sophisticated NLP strategies producing more human-like posts [414].

In this study, the input text was transformed into lowercase and a light stemmer [150] was applied (removing the final "-s") (see Sect. 5.1). A token is defined firstly as a sequence of letters. As a second form, a token could be a sequence of digits and punctuation symbols (e.g., €15.6). Third, punctuation symbols or sequences of them have been kept as is (e.g., !!!). For example, from the input string "Paul's books cost $52.5 !!! #Book http://www.bit.ly/cfGD @Cathy 😍 XXX," the generated tokens are {paul ' s book cost $52.5 !!! # book urllink @cathy 😍 xxx}. In this example, hyperlinks, when present, are replaced by a constant string "urllink" (see Sect. 5.2). For the mentions (e.g., @Cathy) or hashtags (e.g., #Book), the generated tokens are the symbol of the corresponding function followed by the text string in lowercase.

After performing the tokenization of the samples, Table 9.3 reports the number of word-tokens per document[3] (defined as a set of 100 tweets) together with the associated standard deviation in parentheses. The mean values are relatively similar among the different categories, with maybe a value slightly higher for bots and women. The real difference between bots and humans appears when inspecting the standard deviation. This value is clearly higher with bots, signaling a large length variability among the bot's messages.

**Table 9.3** Mean number of tokens (and standard deviation) per set of 100 tweets

| | Bots | Humans | Male | Female |
|---|---|---|---|---|
| Training | 2224.6 (1253.6) | 2069.6 (516.8) | 2015.1 (499.7) | 2123.4 (527.8) |
| Test | 2241.2 (1241.4) | 2031.8 (527.9) | 2030.9 (540.6) | 2033.5 (515.3) |

As another quantitative analysis of this sample of documents, Table 9.4 indicates the size of the vocabulary under different conditions. Under the column "All," the vocabulary size is computed according to the different categories with all possible word-types. Clearly, the humans present a larger vocabulary size compared to the bots (162,452 vs. 101,993). As both classes contain the same number of documents, one can infer that bots must reuse the same words more often than the human category.

When considering only word-types appearing three times or more, the vocabulary size could be reduced by around 50% according to the Zipf's law (Sect. 2.3 shows this effect with the *Federalist Papers*). With tweets and as depicted in column "$tf \geq 3$" (i.e., counting only terms with a frequency larger than two), the size reduction is significantly more important, around 70%. Tweet data includes a large proportion of terms appearing only once or twice, suggesting the presence of more names, spelling variations, and errors in this form of writing.

To reduce the vocabulary size further, one can also ignore words occurring in less than 10 documents. Under the column "$df \geq 10$" ($df$ for document frequency), the vocabulary reduction is more important, with a decrease between 87% for the bots and 91% for the human samples. In the last column of Table 9.4, one can combine the filters performed by the term frequency ($tf$) and the document frequency ($df$) to achieve a smaller vocabulary size and to focus the decision on the most frequent and ubiquitous word-types.

Finally, as an evaluation measure, the accuracy rate (see Eq. 4.1) has been selected. This value varies from 0 to 1 (or 100%); the higher the value, the better the effectiveness of the classifier.

[3]The mean number of word-tokens per tweet can be obtained by dividing the values of Table 9.3 by 100.

**Table 9.4** Vocabulary size after applying different filters

| | Vocabulary size | | | |
|---|---|---|---|---|
| | All (100%) | $tf \geq 3$ | $df \geq 10$ | $tf \geq 20$ & $df \geq 10$ |
| Bot train | 101,993 | 33,889 (33.2%) | 13,260 (13.0%) | 10,126 (9.9%) |
| Human train | 162,452 | 49,340 (30.4%) | 14,732 (9.1%) | 10,285 (6.3%) |
| Male train | 95,412 | 30,394 (31.8%) | 9129 (9.6%) | 6223 (6.5%) |
| Female train | 102,696 | 30,537 (29.7%) | 9227 (9.0%) | 6344 (6.2%) |
| Bot test | 86,406 | 28,658 (33.2%) | 10,362 (12.0%) | 7553 (8.7%) |
| Human test | 117,381 | 36,526 (31.1%) | 10,819 (9.2%) | 7538 (6.4%) |
| Male test | 72,925 | 23,047 (31.6%) | 6666 (9.1%) | 4617 (6.3%) |
| Female test | 71,138 | 22,064 (31.0%) | 6565 (9.2%) | 4420 (6.2%) |

## 9.2 Bots versus Humans

As the bot category covers different types and writing styles, a first solution to identify a set of tweets generated by a machine is to compute the Type–Token Ratio (TTR) (see Sect. 2.4). This measure is the ratio between the number of word-types (vocabulary size) and the number of tokens (or text length) (see Eq. 2.3). When assuming that bots will repeat the same message or a post based on a predefined textual pattern, the same words and expressions will appear many times. In this case, the resulting TTR values will be low compared to a set of tweets written by humans. As shown in Fig. 9.1 based on the training sample, the TTR achieved by bots are usually smaller (mean: 0.36, standard deviation: 0.15) than those obtained by humans (mean: 0.44, standard deviation: 0.05). For example, one can count 422 instances generated by bots having a TTR value smaller than 0.25 vs. 13 for humans.

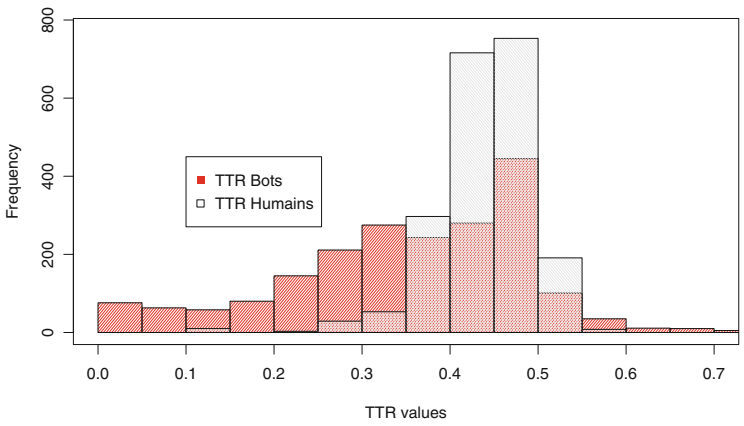


**Fig. 9.1** Distribution of the TTR values for tweets sent by bots or humans

As a second overall measurement, one can compute the lexical density (LD) of a set of tweets defined as the proportion of nouns, verbs, adjectives, and adverbs (see Sect. 2.5 and defined by Eq. 2.10). As template bots generated simple messages or sent simple slogans (e.g., "Make American Great Again" or MAGA), the resulting text will not contain many determiners, pronouns, or prepositions. In such cases, the lexical density will be higher compared to more usual messages in which more functional words will occur.

Figure 9.2 confirms this hypothesis by depicting the LD distribution for both bots and humans. The mean value for bots is 0.64 (standard deviation: 0.14) while for humans the average is 0.56 (standard deviation: 0.05). As for the TTR values, the underlying variability of the data is higher for bots than for humans. For example, one can observe 732 sets of tweets generated by bots having an LD value larger than 0.74 with only 21 instances written by humans.

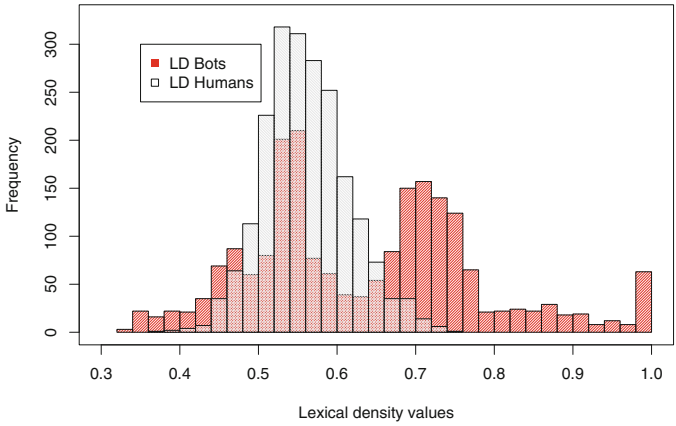


**Fig. 9.2** Distribution of the lexical density values for tweets sent by bots or humans

Based on these results, one can combine the TTR and LD values to propose a filter for detecting the presence of a bot. For example, one can count 917 instances (over 2060) of bots having either a TTR value smaller than 0.25 or an LD larger than 0.7. However, only 33 sets of tweets written by humans will be considered as generated by bots. When applying these thresholds with the test sample, this technique detects 687 instances respecting one of the two constraints in which 643 have been sent by bots. The accuracy rate is 643/687 = 0.936 or 93.6%. Of course, with a test set containing 2640 instances, one needs to take the decision for the remaining part ($2640 - 687 = 1953$ documents). More work is required!

Before focusing on the tweet textual content, one can use the number of hyperlinks, mentions, hashtags, emojis, and emojis with a face. Assuming that the bot's objective is to spread information, one can assume that they will present more hyperlinks and hashtags. One can also hypothesize that they will also include a few mentions as references to similar Twitter accounts. For the emojis, one

could advance the hypothesis that bots will not employ emojis with faces, usually depicting an emotion as indicated by [115]. In Table 9.5, the mean number (and standard deviation) for these feature types are reported.

**Table 9.5** Mean number (and standard deviation) of different features in the bot and human categories (training sample only)

| | Link | Mention | Hashtag | Emoji | Face |
|---|---|---|---|---|---|
| Bot | 85.5 (72.2) | 15.7 (40.1) | 47.5 (131.7) | 33.1 (372.3) | 0.5 (2.9) |
| Human | 44.4 (26.7) | 104.1 (50.0) | 35.6 (43.8) | 29.7 (42.7) | 14.5 (25.7) |

As reported in Table 9.5, tweets generated by bots include more hyperlinks and hashtags. However, for the latter one, the standard deviation is high (131.7), signaling a clear difference between bots. In contrast, humans write tweets with more mentions and more emojis with a face. For the emojis, a larger mean can be found for the bots, but the associated standard deviation is also high indicating a large variability in using this specific feature.

As a first solution to discriminate between bots and humans, an SVM model is built using as predictors the five variables shown in Table 9.5, plus the TTR and LD values. Applying a linear kernel with cost $c = 1$, the system achieves an accuracy rate of 90.15% (2380 correct answers over 2640 instances).

As a second SVM model, the radial kernel has been applied ($cost = 1, gamma = 1$). The resulting effectiveness is very similar to the first model with an accuracy rate of 90.83% (2398 correct answers). From these two models, one can investigate variants with different values for the cost or gamma variables, or by applying other kernel functions such as the polynomial one.

Using the same seven predictors, a logistic regression was generated. The achieved accuracy rate is very similar (0.8973), obtained by producing 2369 correct answers over 2640 instances. To have a better view of the importance of each of these attributes, the coefficient values of the logistic regression are depicted in Table 9.6.

**Table 9.6** Estimates, standard errors, and test for the seven predictors of the logistic regression model

| | Estimate | Std. Error | z value | $Pr(>\|z\|)$ |
|---|---|---|---|---|
| (Intercept) | -2.5973e-02 | 0.50817 | -0.0511 | 9.5924e-01 |
| TTR | 1.3491e+01 | 0.99813 | 13.5165 | 1.2494e-41 |
| LD | -1.6316e+01 | 1.29531 | -12.5964 | 2.2102e-36 |
| Link | 1.7854e-02 | 0.00204 | 8.7564 | 2.0161e-18 |
| Mention | 4.1148e-02 | 0.00148 | 27.7617 | 1.2578e-169 |
| Hashtag | 1.4908e-04 | 0.00102 | 0.1461 | 8.8381e-01 |
| Emoji | -1.5189e-03 | 0.00051 | -2.9903 | 2.7865e-03 |
| Face | 1.1190e-01 | 0.01205 | 9.2860 | 1.6026e-20 |

As one can see, the $p$-values reported in the last column ($Pr(>|z|)$) indicate that the attribute `Hashtag` could be ignored because its coefficient could have a null value (with a significance level $\alpha = 5\%$). Even though the `intercept` is in the same situation, we suggest keeping it. The achieved accuracy rate for a model ignoring the `Hashtag` predictor is the same as the first model (0.8973, with 2369 correct answers over 2640).

During the CLEF PAN 2019 task [314], the best accuracy for this task was 95.95% achieved by [186]. This solution is based on the number of links, mentions, retweets, lowercase letters, and the 800 most important words and bigrams according to the $tf\ idf$ weighing scheme (see Eq. 5.13). As classifier, a logistic regression, and a random forest model [105, 179] were used.

It is clear that to improve our effectiveness level, one must also consider words distinctively used by the bots and humans as well as word $n$-grams or even letter $n$-grams. As this question reappeared when trying to automatically identify the author's gender, the next section is dedicated to expose strategies able to identify terms more related to one of these two categories.

## 9.3 Man vs. Woman

Different studies [97, 428] have indicated that men and women adopt distinct writing styles and favor different words or expressions. According to Pennebaker [299], women tend to employ more personal pronouns, verbs (including auxiliary forms), negations, cognitive words, social words, or emotional terms. Men use more big words (i.e., composed of more than five letters), nouns, determiners, prepositions, numbers, and swear expressions. The frequencies however depend on the communication mode (e.g., pronouns being more frequent in oral). Moreover, each individual presents a more or less strong masculine or feminine persona, and, as expected, some topical words can characterize each gender [13, 216]. In this partition, the LGBT people are not specifically taken into account due mainly to the difficulty to generate a test collection including those characteristics.

The automatic detection of gender on Twitter data was less studied (e.g., see [44, 321]), and the CLEF PAN 2019 evaluation campaign [314] had the goal to fill this gap. Thus, the corpus presented in the first section was reused but considering only the distinction between men and women (see statistics in Tables 9.3 and 9.4). The sub-corpus used in this section contains 3380 documents (set of 100 tweets, see Table 9.2) with the same number of instances (1690) written by men or women.

The best performance to recognize whether a set of tweets was written by a male or a female was 84.32%. The underlying system is based on a combination of word and letter $n$-grams using a logistic regression classifier [133]. From a statistical point of view, the two closest performances (84.17% [17] or 84.13% [109]) could be considered as achieving the same effectiveness level.

Instead of describing the best possible classification model,[4] the presentation will focus on different techniques able to determine the most pertinent features useful to discriminate between a set of tweets written by a man or a woman. In other words, how can we recognize the masculine or feminine aspects in a tweet?

As a first approach, one can assume that women are more talkative than men. Therefore, the tweet size could be a pertinent indicator of the author's gender.[5] As shown in Table 9.3, the mean length (number of tokens) is 2015.1 for men and 2123.4, slightly longer, for women (with similar standard deviations). But as reported by Holmes [164], empirical evidences do not support, in general, this assumption. Depending on the context, women could talk more than men, for example, in private or informal interactions in which the objective of the talk is to maintain social relationships. However, men tend to talk more in formal situations where the aim is to inform or persuade the audience.

As a second (and accurate) approach, the term frequency ($tf$) in both classes can be applied. This strategy was proposed by numerous stylometric studies [45, 46, 339]. As a complementary measure, the document frequency ($df$) could also be useful, indicating in how many documents the corresponding term appears.

Table 9.7 reports the top 15 most frequent terms using either the term frequency ($tf$) or the document frequency ($df$) statistics. Inspecting the ranking provided by the $tf$ values, one cannot see noteworthy differences between men and women. Thus, one cannot directly use such wordlists to discriminate between the two genders. For example, the five most frequent terms are similar for both categories without having exactly the same ranks.

One can however note the importance of Twitter-specific features with a high rank for the mentions (@), hyperlinks (urllink), retweets (rt), and hashtags (#). As for the oral form of communication, personal pronouns (e.g., *I*, *it*, *you*) are very frequent as well as some punctuation symbols (. : ,), and of course some functional terms (*the*, *to*, *of*, *is*, *and*, *in*).

As a third approach, one can estimate the occurrence probability for each word-type in the class male and female and compute their difference. Therefore, when a term is used frequently in one class and appears infrequently in the other, the difference in their occurrence probability should detect them, or at least some of them. Equation 9.1 specifies the needed computation, where $tf_{i,M}$ indicates the absolute frequency of the $i$th term in category M (male), and $n_M$ the text length (in number of tokens) of all tweets belonging in class M (and similarly for the class F [female]).

$$D_{tf}(t_i) = p(t_i, M) - p(t_i, F) = \frac{tf_{i,M}}{n_M} - \frac{tf_{i,F}}{n_F} \tag{9.1}$$

[4] One recurrent critique of such evaluation campaigns is the emphasis on the "competition" aspect in which research teams focus only on a single test collection and a unique performance measure.

[5] When giving this problem to my students, this approach usually appears as one of the first solutions to be proposed.

**Table 9.7** Top 15 most frequent words in male and female category

| | Term frequency ($tf$) | | Document frequency ($df$) | |
|---|---|---|---|---|
| Rank | Male | Female | Male | Female |
| 1 | @ | @ | urllink | of |
| 2 | . | urllink | you | a |
| 3 | the | the | s | to |
| 4 | urllink | : | a | the |
| 5 | : | . | the | and |
| 6 | to | # | is | you |
| 7 | a | rt | it | s |
| 8 | rt | to | in | in |
| 9 | # | , | to | , |
| 10 | of | a | @ | urllink |
| 11 | i | i | of | on |
| 12 | and | and | , | for |
| 13 | in | of | for | is |
| 14 | s | ... | : | it |
| 15 | is | in | and | with |

Table 9.8 reports the top 15 terms achieving the largest or smallest $D_{tf}(t_i)$ values. Those terms are therefore more strongly related to either the male or female category.

**Table 9.8** Largest and smallest $D_{tf}(t_i)$ values ($\times$ by 10,000) on Tweets data

| | Male | | Female | |
|---|---|---|---|---|
| Rank | $D_{tf}(t_i)$ | Term | $D_{tf}(t_i)$ | Term |
| 1 | 74.27 | . | −47.32 | # |
| 2 | 44.04 | the | −38.18 | : |
| 3 | 14.06 | a | −37.27 | rt |
| 4 | 13.55 | that | −27.72 | ... |
| 5 | 10.90 | , | −23.95 | i |
| 6 | 10.77 | is | −16.67 | my |
| 7 | 9.61 | it | −16.63 | you |
| 8 | 8.43 | he | −16.19 | ! |
| 9 | 8.39 | " | −10.74 | _ |
| 10 | 8.23 | ? | −10.64 | so |
| 11 | 8.17 | of | −9.64 | me |
| 12 | 8.09 | s | −9.37 | & |
| 13 | 6.09 | they | −9.20 | thank |
| 14 | 5.93 | but | −8.35 | and |
| 15 | 5.25 | game | −7.40 | love |

According to the data depicted in Table 9.8, men are using the determiners *the*, *a*, or *that* more often and some punctuation symbols (. , " ?). In contrast, women tend to employ more frequently hashtags (indicated with the symbol #), retweets (*rt*), and some punctuation symbols (: . . . ! _ &). They are also using more recurrently the personal pronouns *I*, *you*, *me*, and the possessive adjective *my*. Be careful however, men are also using some pronouns (e.g., *he*, *it*).

Instead of sorting the terms according to their differences in the occurrence frequencies, one can count the number of documents in which they appear (or $df$). In fact, a word-type could present a high probability of occurrence because it appears many times in a few documents. Therefore, its discriminative power could be limited. Thus, it is also worth considering its distribution over the documents.

As discussed previously, to discriminate between the two classes, one can compute the difference between the proportion of documents in which a word-type occurs. In Eq. 9.2, $df_{i,M}$ signals the number of documents in the class M in which the $i$th term appears at least once, and $|M|$ represents the number of documents in the class M, and similarly with $df_{i,F}$ and $|F|$ for the category female.

$$D_{df}(t_i) = \frac{df_{i,M}}{|M|} - \frac{df_{i,F}}{|F|} \tag{9.2}$$

Table 9.9 depicts the top 15 most relevant terms according to this measure. Clearly, this strategy reveals more content terms than the $tf$-based approach. For example, the word *game* occurs in 913 sets of tweets sent by men over a total of 3380 documents, corresponding to 919/3380 = 0.27 (or 27%) of the documents. This term also appears in 500 sets of tweets written by women (or a proportion of 500/3380 = 14.7%). Therefore, the difference is $0.27 - 0.147 = 0.122$.

**Table 9.9** Largest and smallest $D_{df}(t_i)$ values on tweets data

| | Male | | Female | |
|---|---|---|---|---|
| Rank | $D_{df}(t_i)$ | Term | $D_{df}(t_i)$ | Term |
| 1 | 0.122 | game | −0.137 | ❤ |
| 2 | 0.098 | player | −0.120 | girl |
| 3 | 0.078 | league | −0.118 | women |
| 4 | 0.072 | fans | −0.108 | her |
| 5 | 0.072 | point | −0.097 | !! |
| 6 | 0.071 | play | −0.085 | happy |
| 7 | 0.069 | football | −0.085 | omg |
| 8 | 0.064 | team | −0.081 | friend |
| 9 | 0.063 | seem | −0.081 | xx |
| 10 | 0.061 | man | −0.080 | excited |
| 11 | 0.060 | win | −0.078 | woman |
| 12 | 0.060 | fan | −0.076 | she |
| 13 | 0.056 | playing | −0.074 | life |
| 14 | 0.056 | big | −0.074 | beautiful |
| 15 | 0.055 | mate | −0.073 | family |

For the men category, the terms reported in Table 9.9 are clearly related to sports while for women, one can observe several forms related to emotions (e.g., ❤, *happy*, *omg* ("oh my god"), *!!*, *excited*), to social relations (e.g., *girl*, *friend*, *xx*, *family*) or terms occurring frequently in dialogue (e.g., personal pronouns such as *her*, *she*). Of course, instead on being limited to the top 15 word-types, one can consider the first 200–500 terms of each category as pertinent words able to identify both classes.

As a fourth strategy, and as presented in Sect. 5.4, one can apply different feature selection functions playing the role of a filter. Each of these functions computes a score of each term according to their occurrence distribution over the two categories. As we do not have a general theory or empirical evidence favoring one feature selection function over the others, we have applied three of them, namely the odds ratio (OR), the chi-square, and the GSS coefficient. Moreover, according to our experiment described in Sect. 5.4, these three functions tend to promote distinct terms.

**Table 9.10** Top 15 terms extracted according to three feature selection functions

| | Odds Ratio | | Chi-square | GSS |
|---|---|---|---|---|
| Rank | Male | Female | Male-Female | Male-Female |
| 1 | ranger | xxxx | ❤ | game |
| 2 | playoff | strawberry | league | player |
| 3 | utd | mam | game | league |
| 4 | mufc | avocado | player | fans |
| 5 | bud | 🌟 | girl | play |
| 6 | spur | marketer | ♀ | point |
| 7 | fifa | doll | women | football |
| 8 | ufc | bf | xxx | man |
| 9 | mourinho | ☆ | xx | team |
| 10 | kane | zara | omg | though |
| 11 | assumin | sis | lad | fan |
| 12 | ios | acc | fans | mate |
| 13 | 74 | annie | fab | might |
| 14 | garylineker | gals | football | he |
| 15 | messi | sibling | makeup | against |

As some of these functions could excessively favor terms appearing in a few documents in one class and never in the second, it is important to impose some frequency constraints before doing the selection. In the current case, terms occurring less than 20 times ($tf$) or in less than 10 sets of tweets ($df$) are ignored. This procedure reduces the vocabulary around 90% (precise values are reported in Table 9.4).

For the OR function, one can discriminate between terms associated with the male or female class as reported in Table 9.10. For men, important words are related to football with team names (*ranger*, *man*, *utd*, or *mufc* [three denotations related

to *Manchester United*]), organizations (*fifa*), personal names (*mourinho*, *messi*, *garylineker*, *kane*). As another topic, one can see other sports (*ufc*, Ultimate Fighting Championship), technology (*ios*), and some drinks (*bud*).

Terms related to social relations or family (e.g., *sibling*, *annie*, *doll*, *bf* (best friend), *sis* (sister), *gals* [girls]) or with affect (e.g., *xxxx*, and emoticons) are related to the women's profile. As other topics, one can mention shopping (*marketer*, *mam*, *zara*). These distinctions between the two genders reappear when looking at the wordlists provided by the chi-square or GSS function. But one must not overrate these findings. For example, even if the word *football* is clearly related to men appearing 745 times in this category, it might also occur in tweets sent by women (250 occurrences in this case).

As a fifth strategy, and instead of analyzing term by term, related words can be regrouped under the same label defined according to their part-of-speech tags (e.g., determiners, personal pronouns, numbers) or their meaning (e.g., a family wordlist would include {mom, sister, sis, etc.}). For example, the system LIWC [389] proposes such wordlists and other examples can be found with the LSD (Lexicoder Sentiment Dictionary) system [427].

Based on such wordlists, Table 9.11 indicates the frequency per 10,000 terms appearing in tweets sent by men or women. In the last column, the frequency differences between the two genders are reported. The punctuation symbols present the largest difference (58.6) and men tend to employ them more frequently as indicated in bold in the table. Of course, and as reported in Table 9.8, women could use some of them with a higher frequency.

As stylistic features more related to the male category, one can mention the determiners (*the*, *an*), relatives (*ago*, *into*, *both*, etc.), terms related to space (*far*, *over*, *world*, etc.), impersonal pronouns (*that*, *what*, etc.), prepositions (*of*, *in*, etc.), auxiliary verbs (*is*, *be*, *would*, etc.), numbers, and negations (*not*, *never*). Contrary to previous studies, swear words are not so strongly related to the male class.

Table 9.11 also indicates that social words (*lady*, *help*, *give*, etc.) are strongly related to the feminine category as well as emotional terms (*crying*, *hope*, *hate*, etc.) and more specifically positive ones (*love*, *thank*, *care*, etc.). For the negative emotional words, no real difference between the two genders can be found. It is interesting to note that positive emotional words appear four times more frequently than negative ones (the world is usually viewed in a positive way). In addition, one can characterize women's writing styles with the frequent use of biological words (*life*, *heart*, etc.). The characterized lexical features for each gender are depicted in bold in the following tables.

The same kind of analysis was performed but this time focusing on Twitter features and personal pronouns. The frequencies per 10,000 words according to the two genders appear in Table 9.12.

When analyzing the distribution of the personal pronouns, one can perceive distinctions compared to previous studies [13, 299]. First with the personal pronouns, some of them are clearly more associated with men (*he/him*, *they*) and others with females (*I/me/mine*, *you/your*, *we/us/our*). Therefore, for this part of speech, the distinction between genders is more fine-grained in Twitter. Moreover, one can

**Table 9.11** Frequencies per 10,000 words on Tweets data (mean and standard deviation)

| | Male | Female | Difference |
|---|---|---|---|
| Punctuations | **944.8** (104.9) | 886.3 (94.2) | 58.6 |
| Determiners | **435.4** (42.4) | 383.3 (37.9) | 52.1 |
| Relative | **705.1** (56.0) | 677.6 (59.3) | 27.4 |
| Space | **368.6** (33.2) | 343.4 (37.1) | 25.2 |
| Imper. pronoun | **271.5** (38.2) | 247.2 (33.4) | 24.3 |
| Prepositions | **837.2** (59.2) | 813.2 (68.2) | 24.0 |
| Auxiliary verbs | **407.1** (45.3) | 385.9 (43.8) | 21.1 |
| Numbers | **187.1** (41.3) | 166.9 (39.4) | 20.1 |
| Negation | **54.8** (12.8) | 47.7 (11.6) | 7.1 |
| Swear words | 7.6 (5.4) | 5.8 (4.1) | 1.9 |
| Social | 389.1 (54.5) | **430.4** (64.2) | −41.3 |
| Emotion (words) | 222.0 (33.8) | **236.4** (35.9) | −14.4 |
| → Posemo | 179.5 (32.0) | **193.3** (34.5) | −13.8 |
| → Negemo | 42.5 (11.6) | **43.1** (11.7) | −0.6 |
| Biological | 38.1 (10.4) | **5.8** (13.1) | −7.7 |
| Conjunctions | 280.1 (34.6) | 283.0 (37.5) | −2.9 |
| Sexual | 15.9 (7.4) | 22.7 (8.9) | −6.8 |
| Home | 12.4 (5.6) | 15.2 (6.2) | −2.8 |
| Family | 8.2 (4.8) | 11.5 (4.6) | −3.3 |

**Table 9.12** Frequencies per 10,000 words on Tweets data (mean and standard deviation)

| | Male | Female | Difference |
|---|---|---|---|
| Hashtags | 150.5 (84.8) | **193.5** (92.6) | 43.0 |
| Retweets | 152.2 (47.6) | **186.0** (53.7) | 33.8 |
| Emojis | 54.7 (36.5) | **92.5** (52.6) | 37.8 |
| > Emojis emotion | 17.0 (11.9) | **31.0** (20.6) | 13.9 |
| > Emojis face | 11.6 (8.4) | **21.9** (14.8) | 10.3 |
| Mentions | 497.1 (92.6) | **504.1** (100.4) | 7.0 |
| Links | 221.7 (76.1) | 221.9 (54.2) | 0.2 |
| Personal pronouns | 423.6 (67.6) | **487.7** (92.0) | 64.0 |
| Self (I, me, mine) | 172.2 (46.8) | **218.7** (69.3) | 46.5 |
| You/your | 104.6 (27.1) | **127.0** (36.1) | 22.4 |
| We/us/our | 54.8 (16.1) | **62.2** (21.5) | 7.5 |
| She/her | 12.2 (6.5)) | **20.9** (9.7) | 8.7 |
| He/him | **44.1** (17.7) | 31.4 (13.4) | 12.7 |
| They | **33.6** (10.1) | 28.3 (9.5) | 5.3 |

observe a clear distinction with the Self category (*I/me/mine/my*) occurring clearly more often with females than males (see Table 9.12).

In the usage of emojis, one can also detect gender differences. Their frequency of occurrence is higher in tweets sent by women (92.5/10,000 tokens vs. 54.7 for men). Inspecting the length and the different emoji types, men write 😂😂😂, 😂😂, 👍, ⚽, 🏆, 🍻 more frequently while women prefer using 😍, ❤️, 😘, 😊, 📷, 🌊🌊, 🐴, 🐱, 👏. For some emojis, the occurrence frequencies are similar in both genders (e.g., 😂, ❤️). The most frequent emoji[6] appearing on the Internet is 😂 (9.9% of them), and the second is ❤️ corresponding to 6.6%.

As a final analysis, one can focus on letter *n*-grams, starting with unigrams (or single letters), and continuing with letter bigrams or trigrams. As an additional character, the underscore symbol (_) appears to denote the word boundary. In Table 9.13, the most frequent letter *n*-grams are reported according to their length and author's gender. In this presentation, the sequences containing only punctuation symbols are ignored (e.g., ._ or . . . _).

**Table 9.13** Most frequent letter *n*-grams according to the author's gender

| | Male | Female |
|---|---|---|
| Unigrams | t b c p g e h s | a m y 0 i l r n |
| Bigrams | _t th e_ he t_ _b on s_ | i_ _y rt yo _m _r da me |
| Trigrams | _th the he_ ._u on_ e_. | _rt t_@ rt_ _yo you day |
| 4-grams | _the the_ ._ur that hat_ | rt_@ _you . . . _rt day_ you_ |

As expected, the interpretation is more complex. One can however emphasize that men are frequently using the word *the* or *that* and therefore, the most frequent letter is "t." At the level of the bigrams or trigrams, one can observe several substrings related to these words such as "th," "_th," "_the," or "hat_." As the second most frequent letter, one can find "b" and the bigram "_b" ("b" at the beginning of a word) related, for example, to the word *but*.

For the women, the most frequent letters are more difficult to interpret with the "a," "m," or "y." Looking however to the frequent bigrams or trigrams, one can discover the recurrent words behind these letters, for example, *you*, *my*, *me*, *woman*, or *happy*. With the trigrams and 4-grams, one can observe various instances related to the retweet feature (e.g., "_r," "_rt," "rt_," ". . . ," "_rt").

Besides these possible explanations, other frequent *n*-grams are clearly more complex to interpret, such as the high frequency of the letter "p" in the male category or the digit "0" for the female one. Moreover, what is behind the trigram "day" or 4-grams "day_"? Today? Monday?

[6] See `www.home.unicode.org/emoji/emoji-frequency`.

## 9.4 Conclusion

Based on a dataset of 338,000 tweets (CLEF PAN 2019), the first classification task must identify whether a set of 100 tweets was generated by a bot or by a human being. To achieve this and being able to explain the proposed assignment, the overall stylometric measure such as lexical density or type–token ratio (TTR) could be applied as a first efficient filter. Tweets sent by bots tend to repeat the same words and expressions resulting in a low TTR. However, their content could be limited to a simple list of adjectives and nouns, or in other cases, some slogans, producing a high lexical density (few determiners, propositions, conjunctions, or modal verbs). To go further, machine-generated tweets contain more hyperlinks and hashtags (e.g., #IBM), while tweets written by human beings include more mentions (@POTUS44) or emojis with a face (e.g., 😳, 😊). The achieved accuracy rate is higher than 90% with a simple classification model based on seven features.

As a second task, the system must determine whether a set of 100 tweets was written by a man or a woman. This task was clearly more challenging than the first one but not impossible. Looking at the stylistic fingerprints present in tweets as well as the specific characteristics of this social media, one can discover that female writers tend to retweet more often, include more hashtags and emojis, and more first-person pronouns (*I*, *me*, *my*). Taking account of the topics through the usage of different feature selection functions (e.g., odds ratio, chi-square), women write using more terms related to emotions (e.g., ❤️, *happy*, *omg*, *!!*, *excited*), social relations (e.g., *girl*, *friend*, *xx*, *family*), or occurring frequently in dialogue (e.g., some personal pronouns such as *her*, *she*).

Men prefer using some specific pronouns more (*he*, *him*, *they*), including more numbers and negations. Of course, in this communication channel, tweets written by men present more determiners and prepositions (a feature previously identified in other text genres). As male topics, one can find sports, games, some technology (*ios*), and drinks (*bud*).

# Chapter 10
# Applications to Political Speeches


As a third application, the focus is on political speeches and more precisely on US governmental messages. This text genre owns some pertinent characteristics as testbeds for different stylometric methods. First, political documents are freely available, easy to access, and without fee or strict copyright. Second, they are usually of high quality, having none or few spelling errors. Exploring Trump's rhetoric through his tweets is certainly an exception to this quality criterion. Third, political messages are usually relatively easy to read and interpret unlike some scientific documentation. Finally, they can also cover a rather long timespan. This last aspect presents a specific interest in the current target application.

In this chapter, our main objective is to study the evolution of the rhetoric and writing style of the American presidencies from 1789 to 2020. In this perspective, rhetoric is defined as the art of effective and persuasive speaking, the way to motivate an audience. As text sources, both the inaugural speeches and their annual *State of the Union* (SOTU) addresses will be analyzed. These messages indicate the intentions and expose the legislative priorities of the Chief of the Executive. Based on this relatively fixed form, our enquiry will identify some stylistic trends over a timespan covering more than 230 years. In addition, the differences between the precedencies will be detected and illustrated.

The rest of this chapter is structured as follows. Section 10.1 exposes an overview of our political corpus and justifies the choice of both the *State of the Union* and the inaugural speeches. Section 10.2 describes the evolution of some overall stylistic measurements, starting with Washington's presidency to Trump's messages. Section 10.3 proposes to draw two distinct maps showing the stylistic proximity between presidencies. Section 10.4 illustrates how one can detect and extract words, expressions, and sentences that can characterize a presidency. Section 10.5 shows how one can apply different wordlists to measure some linguistics variables and how to combine them to propose some more general measurements. The last section draws a general conclusion.



J. Savoy, *Machine Learning Methods for Stylometry*,
https://doi.org/10.1007/978-3-030-53360-1_10

## 10.1 Corpus Selection and Description

The US presidential function has considerably changed from the time of the young republic under the presidency of Washington to Trump. To identify the broad trends of the underlying stylistic evolution, the number of such addresses remains low until the early twentieth century [403]. After 1945, their volume rose sharply reaching an average of one speech per day under Carter's presidency [153] and has remained at this high level up to now. This evolution can be explained by the growing importance of journalists, media, and in particular, television. Beside their number, the content and the style of the presidential addresses has also evolved during the past two centuries. To indicate the key role played nowadays by the governmental speeches, J. Caesar et al. [48] specify that "speaking is governing." The president must convince the Congress and, more importantly, the citizens of his choice and persuade them that the proposed policies are the most appropriate ones [282]. Recently, the recurrent use of social media also fulfills this major role.

All addresses do not however have the same importance, and their type and audience vary. Faced with the impossibility of analyzing them all, various studies have limited their investigation to speeches considered as essential. For the United States, such analyses focus on the annual *State of the Union* (SOTU) allocutions and/or the inaugural addresses uttered during the swearing-in procedure. To supplement this selection, certain studies add some remarks delivered in a crisis context (e.g., address to the nation after the attacks of Sept. 11th, 2001).

To limit the scope of this application, the analysis of the presidential rhetoric and its evolution will be based on the *State of the Union* (SOTU) addresses. Using computer technology, the entire set can be analyzed, which includes 233 speeches given by 43 presidents from Washington (Jan. 8th, 1790) to Trump (Feb. 4th, 2020). This SOTU address is required by the US Constitution (Article II, Section 3) where it is mentioned that the president must provide information to the Congress about the state of the Union and *"measures as he shall judge necessary and expedient."* Such an address provides an analysis of the current situation, indicates the president's priorities, and presents the legislative agenda for the coming year. All of them are available on the Internet (e.g., www.presidency.ucsb.edu or www.millercenter.org), and an annotated version of the twentieth century addresses was published [198].

In addition, the following reasons explain the importance of the *State of the Union* (SOTU) addresses. First, the United States occupies a position of global importance, consequently, the president's vision transcends the interests of a single country. Second, this set of governmental allocutions covers a period spanning more than two centuries allowing us to analyze the evolution of the rhetoric and style. Moreover, they are delivered in a relatively stable institutional context, reducing some factors of variation. Fourth, several of these speeches have outlined significant political positions such as the Monroe Doctrine (1823), the four freedoms (F. D. Roosevelt in 1941), or the war against poverty (L. B. Johnson in 1964). More recently, this allocution was an opportunity to introduce new phrases such as *"axis*

*of evil"* (G. W. Bush in 2002). Several studies describe in detail the institutional and political context of these speeches, see, for example, [160, 212, 357, 358].

As a second major source of US government speeches, some studies have considered the 58 inaugural addresses uttered at the beginning of each term by each of the 40 elected presidents. This set begins with the first allocution (Apr. 30th, 1789) uttered by Washington and ends with the allocution of Trump (Jan. 20th, 2017). For all of them, the oral form was chosen and the general form and topics are more diverse than with the SOTU speeches. They often present the main objectives for their term in the White House, expose their intentions regarding foreign policy, and give broad guidelines fixed for the new administration. However, their lengths show some variability. For example, the second inaugural speech of Washington (Mar. 4th, 1793) includes only four sentences (145 words) while that of W. Harrison (Mar. 4th, 1841) was the longest with 8356 tokens. A complete list of the selected speeches is provided in Appendix A.6.

One can consider that (oral) speeches delivered by the presidents correspond to an oral communication form while (written) messages (e.g., sent to the Congress) must be categorized as a distinct text genre. However, as mentioned by Biber and Conrad:

> "Language that has its source in writing but performed in speech does not necessarily follow the generalization (written vs. oral). That is, a person reading a written text aloud will produce speech that has the linguistic characteristics of the written text. Similarly, written texts can be memorized and then spoken." [31, p. 262]

If one considers the president as the author of a speech, one does not take this literally. It is known that behind each important politician one can usually find one or usually a team of ghostwriters [176]. For example, under Kennedy's presidency, the main ghostwriter was Sorensen, Madison and Hamilton behind Washington, Favreau[1] and Keenan with Obama, or Scavino behind numerous Trump's tweets. However, though some presidents were actually the author of their speeches (e.g., Lincoln), the tenant of the White House is involved more or less intensively in drafting their important speeches [176]. For example, the website www.youtube.com[2] provides some videos showing the preparation of some of Obama's SOTU addresses.

In our corpus, when two presidents have the same family name, we must be able to distinguish between the two persons. Therefore, we denote *HBush* for the father (George H. W. Bush) and simply *Bush* for his son (George W. Bush). The name

[1] J. Favreau comments on his ghostwriter job at www.youtube.com/watch?v=zFbaesLEa4g.

[2] See at www.youtube.com/watch?v=FxwcJx0-21E the behind the scenes of the *State of the Union* address of 2012, or at www.youtube.com/watch?v=BaR2jXboVQ0 for the SOTU 2014.

*Roosevelt* is reserved for Franklin D. Roosevelt (1933–1945) and by *TRoosevelt* we mean Theodore Roosevelt (1901–1909). The name *Johnson* signals Lyndon B. Johnson (1963–1969) while *AJohnson* corresponds to Andrew Johnson (1865–1869). The name *Adams* designates the second president John Adams (1797–1801) while his son is indicated by *QAdams* (John Quincy Adams, 1825–1829). The name *Harrison* refers to Benjamin Harrison (1889–1893) while *WHarrison* indicates W. Harrison (1841).

Finally, when the analysis is limited to the *State of the Union* addresses, two tenants of the White House (W. Harrison (1841) and J. Garfield (1881)) will be ignored because they just uttered one inaugural allocution, without any SOTU addresses. In fact, their terms were limited to a few months. Moreover, we must recall that our corpus is based on written messages and not on the oral form (e.g., TV interviews, press conferences) or tweets that are often utilized to explain Trump's style. It was demonstrated that for Trump's presidency a rather large stylistic difference does exist between the oral and written form. This clear signal tends to indicate that President Trump did not closely supervise his speechwriters when they wrote his official discourses [345, 348].

## 10.2 Overall Measurements

To analyze the rhetoric and style of presidential writings, a first quantitative measurement focuses on word usages and their frequencies [15, 182, 303]. As the English language has a relatively simple morphology, working on inflected forms (e.g., we, us, ours, or wars, war) or lemmas (dictionary entries) often leads to similar conclusions. In this chapter, the statistics are usually computed based on lemmas.

Simple frequency analysis may report interesting aspects of the evolution of presidential style. For example, Fig. 10.1 illustrates the evolution of the relative frequency of the definite determiner *the* and the lemma *we* using both the *State of the Union* and inaugural speeches. Until Taft's presidency, some stability can be observed with a maximum for the determiner reached by Q. Adams (10.04%). For the pronoun *we*, a maximum is reached under Carter's presidency (4.68%), and this frequency remains relatively high and stable after this extreme value.

As depicted in Fig. 10.1, the pronoun *we* can be more associated with Truman, Carter, Clinton, and Obama. This feature however corresponds to a style-marker for all presidencies after the Second World War. They tend toward a more conversational rhetoric promoting an intimacy between the speaker and the audience [243]. For a political leader, the *we*-words also own the advantage of being ambiguous. What is behind a *we*? The president and the cabinet, the president and the Congress? Often it is a form including the listeners, to create a direct contact with the people as in "you and me," together.

Another simple study may focus on the mean sentence length (MSL) (computed in the number of tokens). The presence of long sentences indicates a substantiated reasoning or specifies the presence of detailed explanations. Usually, a longer

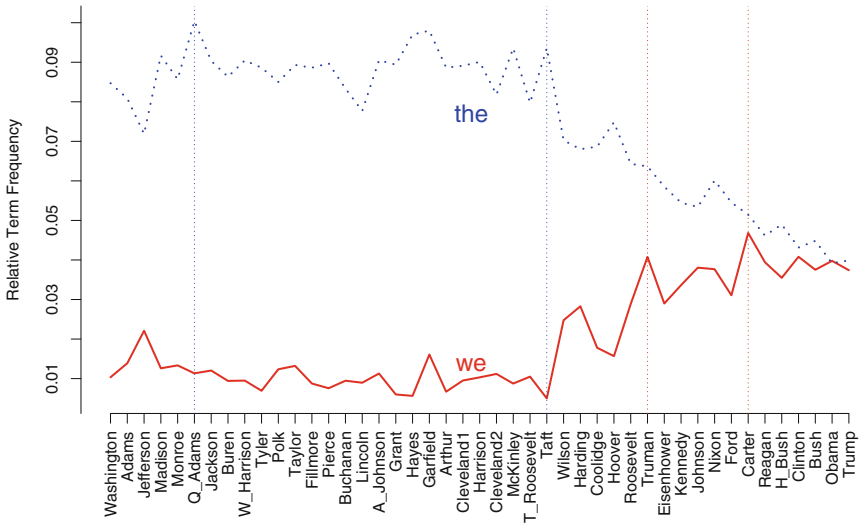


**Fig. 10.1** Evolution of the relative occurrence frequency of the lemmas *the* and *we*

sentence is more complex to understand, especially in the oral communication form. Using the *State of the Union* addresses given by the Founding Fathers, this average value is 40.6 tokens/sentence while the mean over all presidents is 30.6 (some examples are depicted in Table 10.1). With H. Bush (father), the mean sentence length decreases to 16.7 tokens/sentence. These examples, together with Fig. 10.2, clearly indicate that the style is changing over time. Currently, the preference goes to a shorter formulation, more direct, and simpler to understand for the audience.

**Table 10.1** Overall stylistic measurements for some selected presidencies

| | Relative frequency | | | |
|---|---|---|---|---|
| | *the* | *we* | BW | MSL |
| G. Washington | 8.47% | 1.03% | 32.9% | 39.2 |
| J. Madison | 9.17% | 1.26% | 33.0% | 45.0 |
| A. Lincoln | 7.77% | 0.89% | 30.6% | 29.0 |
| T. Roosevelt | 7.97% | 1.05% | 31.2% | 31.5 |
| W. Wilson | 7.03% | 2.48% | 28.0% | 31.2 |
| F.D. Roosevelt | 6.43% | 2.89% | 29.2% | 23.8 |
| W. Eisenhower | 5.85% | 2.90% | 36.1% | 21.3 |
| J. Kennedy | 5.45% | 3.36% | 31.1% | 25.2 |
| R. Reagan | 4.62% | 3.94% | 29.0% | 20.2 |
| H. Bush (father) | 4.89% | 3.55% | 25.9% | 16.7 |
| B. Obama | 3.92% | 3.98% | 26.5% | 19.0 |
| D. Trump | 3.99% | 3.74% | 30.5% | 16.9 |

Word length is another indicator of a message's complexity [153, 389], the longer the words, the higher the complexity. Of course, a simple count of the number of letters to indicate the word complexity should be taken with caution. The letters are not the direct constituent of the word (a role played by the syllables or the morphemes). Moreover, the graphophonic relationship is not direct and simple in the English language. Nevertheless, Lakoff and Wehling's study [232] indicates a relationship between the word length and word complexity analyzed by the receiver.

> "One finding of cognitive science is that words have the most powerful effect on our minds when they are simple. The technical term is basic level. Basic-level words tend to be short. ... Basic-level words are easily remembered; those messages will be best recalled that use basic-level language." [232, p. 41].

Based on this finding, the frequency of word-types composed of six letters or more indicates the use of a rich and sophisticated vocabulary [389]. Of course, this limit of six letters is arbitrary, and another value can be used with an alphabet-based language. For the Chinese language based on sinograms, or for the Korean based on a syllabic system (called Hangul), this limit can be fixed at one or two characters.[3] In fact, Lee et al. [238] found that more than 80% of Korean nouns were composed of one or two Hangul characters, and for Chinese, Sproat [371] reported a similar finding.

A text or a dialogue with a high percentage of big words tends to be more complex to understand. With this measurement, Eisenhower achieves the highest value (36.1%) over all presidents while H. Bush presents the lowest (25.9%) (values are depicted in Table 10.1). As a general tendency, one can see that recent presidencies (from Reagan) tend to employ less big words, in an attempt to simplify their formulations and avoid complex explanations.

A text or a dialogue with a high percentage of big words tends to be more complex to understand. L. B. Johnson recognized this rhetoric problem by specifying to his ghostwriters: "I want four-letter words, and I want four sentences to the paragraph" [281].

Our two last measurements (i.e., mean sentence length and word length) are not fully independent and a relationship does exist between them, known as Arens' law [140]. However, the words are not the direct constituents of a sentence but only through phrases or clauses. Therefore, the relationship is less direct than expected.

Figure 10.2 depicts these first two stylistic measurements with the percentage of big words in the addresses delivered by all presidents (x-axis) and the mean sentence

[3] Instead of directly considering the sinogram or the Hangul character, one can count the number of strokes required to draw the corresponding character.

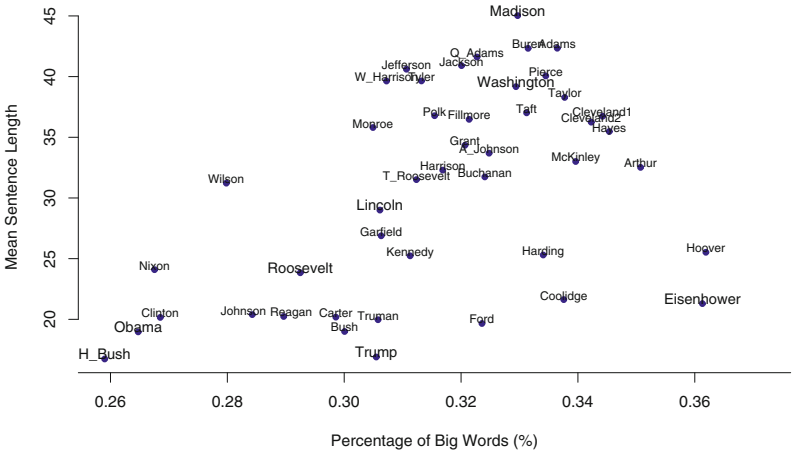


**Fig. 10.2** Relationship between percentage of big words (BW) (longer than five letters, x-axis) vs. mean sentence length (MSL, y-axis) based on the *State of the Union* and inaugural addresses

length (y-axis, computed according to the number of tokens). On the top, one can find the Founding Fathers (with Madison depicting the highest mean sentence length (45.0 tokens/sentence)), and below them the presidents of the nineteenth century. An interesting finding in this graph is the position of Lincoln (1861–1865) and Roosevelt (1933–1945) depicting a lower mean sentence length and a smaller percentage of big words than their predecessors (Wilson is an exception for Roosevelt). For their contemporaries, their styles appeared as clearly distinct compared to previous presidencies.

On the bottom left, we see the contemporary presidents (H. Bush, Obama, and Clinton) opting for short sentences and few big words. On the extreme right with a large percentage of big words, we discover Hoover (1929–1933) and Eisenhower (1953–1961) having the largest percentage of big words (36.1%). Trump presents the second smallest mean sentence length (16.9 tokens/sentence), slightly more than H. Bush (16.6).

## 10.3 Stylistic Similarities Between Presidencies

To visualize the stylistic similarities and differences between the US presidencies, the distribution of the part-of-speech (POS) categories can be chosen as overall stylistic markers. To achieve this, a presidential profile is generated by concatenating all their speeches. Then to generate a stylistic surrogate, all POS tags have been selected, including two additional categories: one for names, and the other for the periods. With this last item, texts depicting long sentences can be distinguished from those containing short ones.

Based on the relative occurrence frequency of each POS category, Obama is the US president using verbs (16.4% of his tokens) and adverbs (5.2%) more frequently. This aspect indicates a speech oriented more toward action (real or proposed). The highest rate of nouns can be found with Hoover (1929–1933) (23.4%), specifying that the author tries mainly to explain the situation (e.g., the economic downturn in the 1930s). When considering determiners and prepositions, Q. Adams (1825–1829) uses them with the highest frequency (20.1% for the prepositions, 14.1% for the determiners). Eisenhower (1953–1961) opts clearly for adjectives (9.4%), while Clinton (1993–2001) prefers pronouns (9.5%), usually more frequent in dialogues. H. Bush (1989–1993) employs the full stop most frequently (5.4%), indicating a bias in favor of short sentences (this aspect was already identified in Fig. 10.2).

Instead of limiting the analysis on each category separately, one can position each presidency according to their occurrence frequencies of verbs (*verb*), adverbs (*adv*), adjectives (*adj*), nouns (*noun*), names (*name*), pronouns (*pronoun*), determiners (*deter*), prepositions (*prep*), conjunctions (*conj*), and periods (*period*). To achieve this visual representation, the principal component analysis (PCA) [15, 235] was applied and the result is depicted in Fig. 10.3[4] (see Sect. 3.5). This representation generates two orthogonal composite components taking into account $45.7\% + 19.1\% = 64.8\%$ of the total underlying variability.

When analyzing this figure, the horizontal axis indicates the opposition between the frequent use of determiners and prepositions on the left, and pronouns, periods, and adverbs shown in the right part. On the left, one can see Q. Adams (1825–1829) (with the largest frequency of prepositions and determiners) and some presidencies covering the end of the nineteenth century and the beginning of the twentieth century such as Taft (1909–1913), Arthur (1881–1885), McKinley (1897–1901), Hayes (1877–1881), Harrison (1889–1893), and as an exception to this rule, Madison (1809–1817) (appearing under Pierce). On the right of Fig. 10.3, we find the latest presidencies with their high frequency of pronouns and shorter sentences (Reagan, Bush, Obama, and Trump).

The vertical axis signals the frequent use of adjectives and nouns (downward direction) while names and verbs are more associated with the upward direction. On the lower part of the figure, one can see Eisenhower (1953–1961), Kennedy (1961–1963), as well as Hoover (1929–1933). In the opposite direction, Monroe's style (1817–1825) appears distinct from the others by using more names and verbs.

In the center of Fig. 10.3, where the two axes are crossing, we encounter presidents having an average use of all POS categories, such as Garfield (1881), Coolidge (1923–1929), or van Buren (1837–1841), and not too far, T. Roosevelt (1901—1909).

As an overview, this figure depicts the first presidencies in the top-left part with Monroe, Jefferson, or Washington. Going clockwise, the upper-right region shows contemporary presidents with Obama, Clinton, Bush, or Reagan. On the bottom right, one can see presidencies of the years 1920–1980 with Wilson, Harding,

[4]This figure was generated by the function `biplot()` in R.

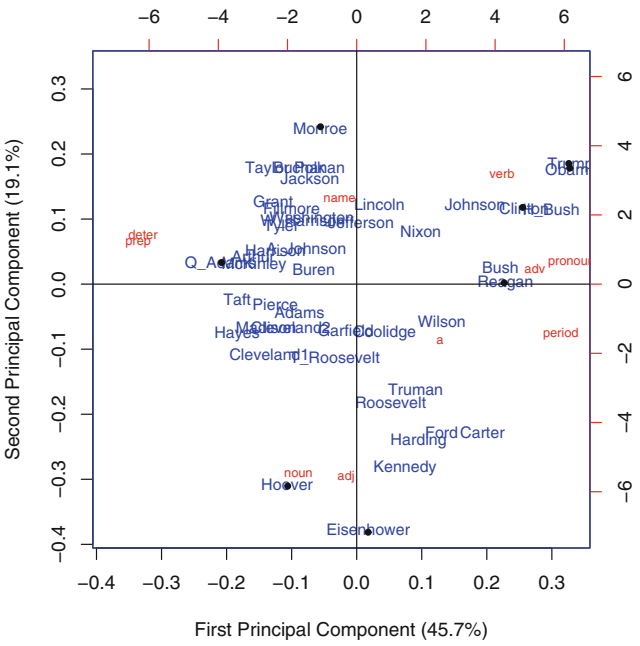


**Fig. 10.3** Representation of each US president according to their usage of the different part-of-speech categories

Roosevelt, Eisenhower, Truman, Ford, and Carter. On the bottom left, one can mainly find the presidential style corresponding to the years 1850–1910 with Pierce, Hayes, Arthur, Cleveland, T. Roosevelt, and Taft.

Instead of considering part-of-speech categories, one can take account of the most frequent word-types or lemmas as stylistic markers. Following this view, each presidential profile (concatenation of all his SOTU speeches) is represented by the top 300 most frequent lemmas occurring in their addresses [340]. To compute the distance between each pair of profiles, the Labbé's intertextual distance has been computed (see Sect. 3.3). With this metric, the returned value depends on the overlapping between the two texts and varies between 0.0 and 1.0. Between these two extremes, the distance depends on the number of lemmas in common in both texts and their frequencies.

Applying this distance measurement for each pair of profiles, a symmetric matrix ($43 \times 43 = 1849$ values) is computed. Just showing all these values does not provide a clear and easy-to-understand picture. This information however provides the input for an automatic classification procedure [15, 201]. The result, depicted in Fig. 10.4, allows us to draw a map generated to reflect as closely as possible the distances between the presidential profiles.

In this graph, the distance between each president is visualized by a technique derived from genomic trees [199], a package available in R (see also [182]). In

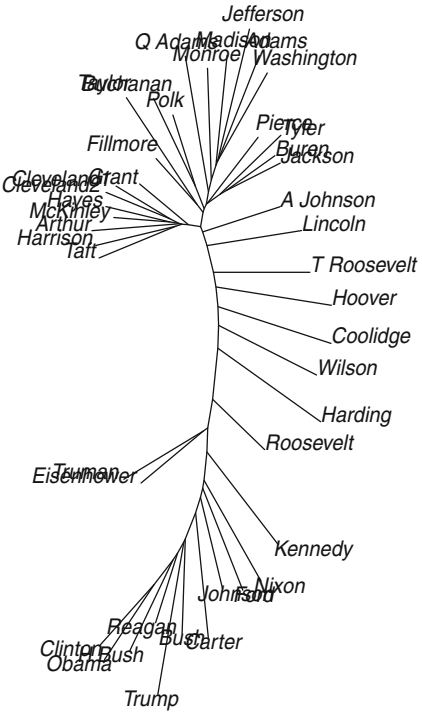


**Fig. 10.4** Tree representation of stylistic distances between the president profiles (with 300 most frequent lemmas) based on the *State of the Union* and inaugural addresses

this picture, the line length joining two presidents is proportional to the distance between them. For example, to go from Lincoln to Kennedy, we must travel a greater distance than between Lincoln and T. Roosevelt. Finally, to be on the left or the right, up or down, does not matter. This position is selected to allow a better overall visualization. In Fig. 10.4, the smallest distance can be found between Jackson and van Buren[5] (0.066) and the largest between Q. Adams and Obama (0.338).

Following a movement from top to bottom, the presidents are placed almost in chronological order. On the top of the figure, the first group includes the Founding Fathers (Washington, Adams, Jefferson, Madison, and Monroe) and also Q. Adams. This group covers the period from 1790 to 1829. A closer inspection reveals that Monroe (1817–1825) and Q. Adams (1825–1829) are a little bit farther apart from the kernel formed by the first four presidents (1798–1817). Although the stylistic distance is still small among these six presidents, the political vision of Jefferson

[5]Van Buren was the vice president during Jackson's second term (1833–1837), a fact that could explain this close stylistic similarity.

or Madison (limited federal power) is different from that shared by Washington and J. Adams (strong federal government).

The second period comprising mainly the years 1829–1861 is located just on the right, formed by the two Democratic duos, first with Pierce (1853–1857)–Tyler (1841–1845), and a second pair with Jackson (1829–1837)–Van Buren (1837–1841). On the left, and closely related, one can find the two pairs, first with Polk (1845–1849)–Buchanan (1857–1861) and then with Taylor (1849–1850)–Fillmore (1850–1853).

Corresponding to the years 1869–1913, the second half of the nineteenth century appears in two groups. On the left, beginning with Grant (1869–1877), one can find a group composed of Republicans with Hayes (1877–1881), McKinley (1897–1901), Arthur (1881–1885), Harrison (1889–1893), and Taft (1909–1913), together with the two terms of the Democratic President Cleveland (1885–1889, 1893–1897).

In the median section, one can find a sequence of presidents having a style significantly different from each other. The first in this series is A. Johnson (1865–1869) followed by Lincoln (1861–1865), who owns a style usually considered as the most beautiful and very distant from his predecessor (Buchanan).

Within the first half of the twentieth century, three presidents clearly stand out from a stylistic point of view. First, T. Roosevelt (1901–1909) depicts a large distance with his predecessor (McKinley) and his direct successor (Taft), both appearing in the group of presidents covering the period 1869–1913.

Second, Wilson (1913–1921) modernizes the presidency; the United States becomes a great power wishing to play a global role [284]. Wilson adopts a clearly distinctive style to help him in achieving this goal. The third innovative president is Roosevelt (1933–1945) whose style stands out clearly from his predecessors (Coolidge and Hoover) both located between T. Roosevelt and Wilson. Harding (1921–1923) also appears as owning a distinct style compared to the others, but his presidency was judged as one of the worst in US history [318].

From Truman (1945–1953) to Reagan (1981–1989), each presidency depicts a pretty distinct style. As an exception, we can find the pair Eisenhower (1953–1961)–Truman (1945–1953). A quick inspection reveals that the presidents appear in almost perfect chronological order; the exception are Nixon (1969–1973) and Ford (1973–1977), appearing just after Kennedy (1961–1963).

Finally, the last group is composed of contemporary presidencies close to the chronological order with Reagan, H. Bush (1989–1993), Clinton (1993–2001), and Obama (2009–2017). From a stylistic point of view, Bush (son) constitutes a step backward (toward Reagan). Trump's presidency (2017–2020) continues in this backward direction depicting distinctive stylistic markers. This presidency is related to both H. Bush (0.131) and Bush's (0.135) presidency.

## 10.4 Characteristics Words and Sentences

To discriminate between several writing styles, the frequency variations of functional words or a subset of them [373] can provide pertinent features. Those terms are however of limited interest when focusing on the semantic level, assuming that authors tend to limit themselves into a subset of possible topics. Thus, each author can be characterized by both some functional and topical words and expressions he likes and uses frequently. For example, the word *florins* or the bigram *British subjects* cannot characterize recent US presidents. But those expressions match the vocabulary of the first presidencies well (e.g., Washington). To determine those overused words or expressions, Muller's method [275] can be applied (see Sect. 5.6). This approach was already suggested to generate an authorship attribution model [336]. But our current interest is to illustrate several applications that can be obtained by this feature extraction method.

First, applying this selection strategy to the SOTU and inaugural speeches, one can identify the isolated words and word $n$-grams characterizing each presidency. Instead of taking account of all 45 presidencies, Table 10.2 reports some overused words when considering only the last 11 presidents, beginning with Kennedy (1961) to Trump (2020).

**Table 10.2** Top ten overused terms of some recent presidencies

| Kennedy | | Nixon | | Bush | | Obama | | Trump | |
|---|---|---|---|---|---|---|---|---|---|
| Score | Term | Score | Term | Score | Term | Score | Term | Score | Term |
| 12.6 | communist | 14.6 | which | 18.8 | Iraq | 14.2 | that | 14.4 | thank |
| 10.3 | alliance | 11.2 | shall | 16.2 | terrorists | 13.0 | why | 13.5 | he |
| 10.1 | of | 9.41 | as | 13.1 | terror | 12.1 | job | 13.1 | very |
| 9.8 | farm | 9.3 | peace | 12.1 | enemy | 11.3 | business | 12.9 | including |
| 8.2 | Atlantic | 9.2 | structure | 11.0 | regime | 9.7 | get | 12.2 | border |
| 7.6 | free | 8.7 | the | 10.2 | and | 8.9 | like | 11.5 | working |
| 7.5 | shall | 8.5 | in | 9.9 | terrorist | 8.7 | do | 11.3 | immigration |
| 7.4 | which | 7.8 | goal | 9.4 | relief | 8.3 | college | 10.9 | incredible |
| 7.3 | west | 7.8 | great | 9.3 | September | 7.8 | how | 10.7 | illegal |
| 7.1 | aid | 7.4 | rather | 8.7 | yet | 7.6 | company | 10.4 | American |

As expected, these lists present both functional and topical terms. With Kennedy's presidency, one can observe functional words such as *of*, *shall*, *which*, and later in the list the terms *the*, and the comma. In addition, each presidency exposes some topical terms related to problems and difficulties encountered by the presidency. As an example, the Bush presidency (son) illustrates this aspect very well (*Iraq*, *terrorists*, *terror*, *enemy*, ...).

Second, one can also see some word bigrams or trigrams such as *balance of payment* (Kennedy), *full employment* (Nixon), *arms reduction* (Reagan), *Al Qaida* or *tax relief* (Bush), *clean energy* (Obama), as well as *health care*, *southern border*

(Trump). Usually such expressions provide a better understanding of the underlying issues.

Third, instated of considering overused terms per presidency, one can select a few of them and analyze their evolution over time. As reported in Table 10.3, these examples are useful to show the presidential stylistic development. One can observe that the overuse of the determiner *the* and the preposition *of* belongs to the 60s' writing style. As another specimen, the full stop characterizes both H. Bush (father) and Trump's presidencies well (see also Fig. 10.2). Another hint indicates that nowadays short sentences are the norm.

**Table 10.3** Some overused terms across the last 11 presidencies

| Term | Presidencies | | | |
|---|---|---|---|---|
| the | Kennedy | Johnson | Nixon | Ford |
| of | Kennedy | Johnson | Nixon | Ford |
| , | Kennedy | Reagan | Trump | |
| . | H Bush | Trump | | |
| we | Ford | | | |
| I | Johnson | Ford | H Bush | Clinton |
| America | Nixon | Bush | Trump | |
| American | Trump | | | |
| God | Reagan | Trump | | |
| *Adjectives* | Kennedy | Ford | Carter | |
| *Names* | Bush | Trump | | |
| *Adverbs* | Obama | Trump | | |

Fourth, and in addition to words, one can also analyze the occurrence frequencies of different part of speech tags. As example, Table 10.3 depicts that more recent presidencies favor proper names (e.g., persons, products, geographical entities) and adverbs (e.g., very) while the 60s were more related to nouns and adjectives.

Finally, to extract more explicit examples of the style and rhetoric of each presidency, one can determine some characteristic sentences. Such a sentence can be defined as one having the largest number of overused terms. But such an operational definition would favor longer sentences. Thus, the sentence score can be defined as the count of the number of overused words divided by the square root of the sentence length. Based on this definition, the computer can reveal some examples for each presidency (more details are given in [335, 343]).

As an interesting first case, one can read three of the most characteristic sentences from Trump's speeches. The words occurring in the list of overused terms are written in italics (as well as the comma and the period). In this case, the underlying issues are clear and correspond to defined priorities of Trump's administration. As one can observe, the emphasis is put on the president's country (*USA*, *US*) or the people (*American*). The recurrent use of *proudly* and *beautiful* corresponds to a

more assertive and idealistic tone [243], an increasing tendency appearing during the last presidencies. And with Trump, one can add a more grandiose tone [6, 154].

> "Our new *U.S.*–Mexico–Canada Agreement – or USMCA – will replace *NAFTA* and deliver for *American* workers: bringing back our manufacturing jobs, expanding *American* agriculture, protecting intellectual property, and ensuring that more cars are *proudly* stamped with four *beautiful* words: made in the *USA*." D. Trump, *State of the Union*, Feb. 5th, 2019.
>
> "In the last 2 years, our *brave ICE officers* made 266,000 arrests of *criminal aliens*, *including* those *charged* or convicted of nearly 100,000 *assaults*, 30,000 *sex* crimes, and 4,000 *killings*." D. Trump, *State of the Union*, Feb. 5th, 2019.
>
> "Before I came into office, if *you showed* up illegally on our *southern border* and were *arrested*, *you* were simply *released* and *allowed* into our *country*, never to be seen again." D. Trump, *State of the Union*, Jan. 25th, 2020.

As a second example, the system extracts the following sentences from Obama's speeches. As reported in Table 10.2, the words *that*, *job*, *business*, *like*, and *do* are overused by the most recent former president. In addition, the negation (*not*) and the semi-colon also appear more frequently.

> "The American people deserve a tax code *that* helps small *businesses* spend less time filling out complicated forms, and more time expanding and hiring; a tax code *that* ensures billionaires with high-powered accountants can*not* pay a lower rate than their hard-working secretaries; a tax code *that* lowers incentives to move jobs overseas, and lowers tax rates for *businesses* and manufacturers *that* create *jobs* right here in America." B. Obama, *State of the Union*, Jan. 25th, 2014.
>
> "It is *because* they understand *that when* I *get* tax breaks I *do not* need and the country can*not afford*, it either adds to the deficit, *or* somebody else has to *make up* the difference, *like* a senior on a *fixed* income*; or* a *student* trying to *get* through school*;* or a family trying to *make* ends meet." B. Obama, *State of the Union*, Jan. 24th, 2012.

As last examples, one can read the following sentences and compare the overused terms with those reported in Table 10.2. Nixon's presidency clearly contrasts with the last two US presidents with the overuse of the determiner *the* and the preposition *of* and the presence of longer sentences.

> "We still have a *lot more to do*, all of us, *to* make *welfare* reform a success; providing *child care*, helping *families* move closer *to* available jobs, *challenging more* companies *to* join our *Welfare to Work* Partnership, increasing *child-support* collections from deadbeat *parents* who have a duty *to support their* own *children*." B. Clinton, State of the Union, Jan. 27th, 1998.
>
> "*Let* us build a *structure of peace in the* world *in which the* weak are *as* safe *as the* strong, *in which each* respects *the* right *of the* other to live by a different system, *in which those* who would influence others will do so by *the* strength *of* their ideas and not by *the* force *of* their arms." R. Nixon, Inaugural speech, Jan. 20th, 1973.

To analyze the vocabulary evolution over time, our examples are based on the diachronic corpus of US presidential messages. However, similar methods and techniques can be applied to identify the main trends over a given domain of knowledge. For example, based on the titles of the *Journal of the American Statistical Association* (JASA) (1888–2012), Trevisani and Tuzzi [400, 401] identify a first period (1888–1920) in which statistics were an instrument of the government, with overused terms such as *census*, *birth rate*, and *labor statistics*. In the second epoch (1920–1960), the articles are focusing on the economy (with the Depression, and the post-war period) with expressions such as *manufactured goods*, *cost of living*, and *price index*. During the years 1960–1990, statistics emerge as an autonomous scientific domain with its own vocabulary (with expressions such as *probability*, *sampling*, *regression*). The last era (1990–2012) can be characterized by larger term variability with recurrent terms such as *algorithms*, *bootstrap*, *smoothing*, *robust*, *networks*, etc. showing the opposition between frequentist and Bayesian data analysis, parametric and non-parametric, or linear and non-linear methods. A similar study on the evolution of the lexicon on humanities and social sciences can be found in [405].

## 10.5 Rhetoric and Style Analysis by Wordlists

Text analysis can also be performed using a set of wordlists, each containing words or expressions related to a given subject, reflecting an attitude or producing a particular tone. For example, in a wordlist entitled *Self*, one can regroup all possible forms of the first-person singular pronouns as well as related possessive adjectives {I, me, myself, mine, my}. Following this example, different wordlists can be

generated corresponding to different closed part of speech (POS) categories, such as personal pronouns, prepositions, conjunctions, determiners, etc. For open POS categories (e.g., nouns, adjectives, verbs), it is impossible to enumerate all possible forms, and one can take account of the most frequent ones.

A deeper analysis could be achieved by establishing more specific wordlists. With the goal to analyze the rhetoric adopted by US political leaders, the Diction system [153, 155] was built with wordlists generated for this domain. For example, the wordlist denoted *Symbolism* regroups words associated with designative and ideological language in US politics with terms such as *America*, *American*, *country*, *nation*, *law*, *people*, *rights*, etc. [153]. When measuring the tenacity of a political leader, this system proposes the *Rigidity* list containing mainly all forms of the verb to be (e.g., are, is, was, will), usually used to denote a complete certainty. With the *Intellectuality* set of words (e.g., believe, chose, consider, know, think), one can estimate the degree of cognition or reflective aspects in a message or during an interview.

Built also on several wordlists, the LIWC system [299, 389] is focused on determining psychological as well as emotional traits of the author (e.g., such as depressed, open-minded, extraverted, etc.). In this case, lists may regroup tokens related to a given topic (e.g., *Human* with (e.g., child*,[6] family, friend), or *Social* with (e.g., societ*, speak, tell, team)), terms denoting an emotion (*Posemo* with (e.g., hope, win, best), *Negemo* with (e.g., fear, tear, sadness)) or other rhetoric aspects such as *Cognitive* mechanism (e.g., cause, think, organize, realis*), or *Tentative* language forms (e.g., maybe, perhaps, appear).

These two systems are not the only ones. Just to mention a few, one can cite the General Inquirer[7] trying to cover all possible aspects in a language [381]. Targeting political texts, the LSD (Lexicoder Sentiment Dictionary, www.lexicoder.com) system is focusing on detecting sentiments or affects in news, party's manifestos, as well as transcripts of interviews or TV debates [427]. Dedicated to detecting opinions and private states (e.g., beliefs, sentiments, speculations, etc.), the MPQA (www.mpqa.cs.pitt.edu) [89] is also developed in a series of components, each dedicated to a specific target application (e.g., to recognize stances in online debates, to identify argumentations).

Pursuing our analysis of political speeches, one can compute the amplitude achieved by the different presidencies over a set of selected wordlists, each reflecting a given aspect. When adopting this approach, one assumes that the resulting frequencies form a reliable measurement of the underlying emotions or tone. As an example, data depicted in Table 10.4 provides an overview of some recent US leaders. These values signal the percentage per thousand of terms belonging to a wordlist when considering both the SOTU and inaugural speeches.

[6]The asterisk (*) indicates the possible presence of any string, for example, "-ren" or "-hood" in our example.

[7]The corresponding website is located at www.wjh.harvard.edu/~inquirer/.

**Table 10.4** Frequencies per thousand of some selected word-types or wordlists

| Measurement | Nixon | Reagan | Clinton | Bush | Obama | Trump |
|---|---|---|---|---|---|---|
| Personal Pronouns (LIWC) | 59.8‰ | 62.5‰ | **74.6**‰ | 64.2‰ | 68.6‰ | 69.5‰ |
| Self (Diction) | 10.8‰ | 9.1‰ | **13.3**‰ | 8.3‰ | 10.9‰ | 9.6‰ |
| Symbolism (Diction) | **28.7**‰ | 24.5‰ | 22.2‰ | 26.8‰ | 18.8‰ | 26.0‰ |
| Posemo (LIWC) | 14.5‰ | 15.6‰ | 15.6‰ | **17.3**‰ | 14.1‰ | 15.6‰ |
| Negemo (LIWC) | 4.0‰ | 3.5‰ | **4.2**‰ | 3.9‰ | **4.2**‰ | 3.5‰ |
| Intellectuality (Diction) | 4.9‰ | 4.8‰ | 5.9‰ | 4.6‰ | **6.1**‰ | 4.3‰ |
| Rigidity (Diction) | 38.5‰ | 38.0‰ | 30.4‰ | 36.7‰ | 37.8‰ | **38.8**‰ |
| Human (Diction) | 55.0‰ | 62.4‰ | **73.6**‰ | 65.1‰ | 63.9‰ | 65.8‰ |
| Tentative (LIWC) | 10.3‰ | 10.4‰ | 10.9‰ | 10.6‰ | **13.7**‰ | 8.3‰ |
| Cognitive (LIWC) | 136.2‰ | 140.9‰ | 153.0‰ | 145.3‰ | **154.7**‰ | 133.2‰ |

Table 10.4 shows that Clinton presents the highest measurement for personal pronouns (values depicted in bold), followed by Trump and Obama. Clinton also exhibits the highest value for the *Self* category. The *Symbolism* category tends to characterize Nixon's presidency, a variable also relatively high during the Bush (son) presidency. The latter also shows a frequent use of positive emotions (*Posemo*). With the class of negative emotions (*Negemo*), the overall values are low for all presidents. A political leader in power does not want to talk with negative feelings, an emotion appearing more during an electoral campaign or from the mouth of the opposition leaders.

Obama's rhetoric is more related to *Intellectuality*, and *Cognitive* terms and expressions. In addition, this president shows the highest degree of *Tentative* words, implying the presence of some nuances in his speeches. However, this measurement exhibits the lowest value with Trump's presidency. For this last one, the *Rigidity* gauge obtains its highest value, leading to the conclusion that certitude is one key aspect of Trump's speeches. Moreover, the cerebral aspect also attains a low point with small values for the *Intellectuality* and *Cognitive* indicators (Trump's election was based on anti-elite and anti-intellectualism consideration [154, 274]). Moreover, compared to the previous presidency, the *Symbolism* is also relatively high and references to the country is also a recurrent characteristic of Trump's presidency. As demonstrated by Hart [154], Trump's presidency can be characterized by either a high or a low value on numerous such linguistics indicators; from a stylistic point of view, Trump is a president of extremes.

As previously described, one can employ each of these wordlists to determine some sentences reflecting the highest values for the underlying indicator for a president. For example, with the *Symbolism* class, the system detects the following sentences from Kennedy's, Bush's, and Trump's presidency.

> "A newly conceived *Peace* Corps is winning friends and helping *people* in fourteen *countries*, supplying trained and dedicated young men and women, to give these new *nations* a hand in building a society, and a glimpse of the best that is in our *country*." J. F. Kennedy, *State of the Union*, Jan. 11th, 1962.
>
> "In Afghanistan, *America*, our 25 NATO allies, and 15 partner *nations* are helping the Afghan *people* defend their *freedom* and rebuild their *country*." G. Bush, *State of the Union*, Jan. 28th, 2008.
>
> "For many decades, we've enriched foreign industry at the expense of *American* industry, subsidized the armies of other *countries* while allowing for the very sad depletion of our military. We've defended other *nations*' borders while refusing to defend our own and spent trillions and trillions of dollars overseas while *America*'s infrastructure has fallen into disrepair and decay." D. Trump, Inaugural speech, Jan. 20th, 2017.

Instead of indicating the amplitude of a dimension by a percentage as reported in Table 10.4, one can verify whether or not a presidency presents a noteworthy higher or lower value. In other words, the underlying question is whether or not a value such as 74.6‰ for personal pronouns under Clinton's presidency (see Table 10.4) is really different from the mean over all presidencies? To achieve this, one can compute the Z score of each value $v_i$ for the $i$th category as shown in Eq. 10.1, where $mean_i$ denotes the average over all presidencies and $sd_i$ the standard deviation.

$$Z\ score(v_i) = \frac{v_i - mean_i}{sd_i} \tag{10.1}$$

The resulting Z score value is a dimensionless quantity signaling the difference compared to the mean. This last value is computed over all selected presidencies. As it makes no sense to match the last presidents with the Founding Fathers, our comparison will be limited to the last 11 presidents, from Kennedy to Trump.

Table 10.5 depicts these Z score values based on the information provided in Table 10.4 with, in addition, the values achieved by the presidents not shown in Table 10.4 (e.g., Kennedy, Johnson, Ford, Carter, and H. Bush).

Assuming that the Z score values follow a Gaussian distribution, one can admit that values between −2.0 and 2.0 reflect a normal situation.[8] One president might use a little bit more or less a given set of terms compared to the mean. However, Z score values larger than 2.0 signal an over-use in the corresponding dimension. In Table 10.5, all reported values are inside the normal limits. One can however

[8]With a standardized Gaussian distribution (mean = 0, standard deviation = 1), 95% of the values appear between −2 and +2.

**Table 10.5** Z values of some selected stylistic and rhetoric indicators

| Measurement | Nixon | Reagan | Clinton | Bush | Obama | Trump |
|---|---|---|---|---|---|---|
| Personal Pronouns (LIWC) | −0.22 | 0.05 | 1.30 | 0.23 | 0.68 | 0.78 |
| Self (Diction) | −0.16 | −0.62 | 0.56 | −0.87 | −0.13 | −0.48 |
| Symbolism (Diction) | 1.68 | 0.61 | 0.02 | 1.19 | −0.86 | 0.99 |
| Posemo (LIWC) | 0.26 | 0.79 | 0.77 | 1.59 | 0.04 | 0.77 |
| Negemo (LIWC) | 0.33 | −0.72 | 0.69 | 0.09 | 0.69 | −0.71 |
| Intellectuality (Diction) | −0.03 | −0.10 | 0.78 | −0.31 | 0.90 | −0.57 |
| Rigidity (Diction) | 0.69 | 0.57 | −1.58 | 0.19 | 0.52 | 0.79 |
| Human (Diction) | −0.24 | 0.45 | 1.49 | 0.70 | 0.58 | 0.76 |
| Tentative (LIWC) | −0.26 | −0.20 | 0.04 | −0.12 | 1.37 | −1.19 |
| Cognitive (LIWC) | −0.32 | −0.08 | 1.09 | 0.45 | 1.24 | −0.57 |

underline that Clinton depicts a high Z score values for the personal pronouns and *Human* category while Nixon presents a high value for the *Symbolism* class. Moreover, the last three Republican presidents in this table (Reagan, Bush, and Trump) share in common a high Z score value in both the *Symbolism* and *Posemo* dimensions.

Finally, these stylistic and rhetoric measurements can be combined to promote new and more general variables [153, 155, 363]. For example, as shown in Chap. 9, the male style can be characterized by the frequent use of articles, prepositions, negations, big words, and swear formulations. In contrast, one can observe infrequent use of some pronouns and social words. Based on these findings, one can suggest a male style variable defined by the following formula:

$$\begin{aligned} Male = {} & Z\ score(Article) + Z\ score(Preposition) + Z\ score(Negation) \\ & + Z\ score(BigWords) + Z\ score(Swear) \\ & - Z\ score(Pronoun) - Z\ score(Social) \end{aligned} \tag{10.2}$$

In this expression, the different measurements are transformed into their Z score values before entering them into the overall measurement. Moreover, as specified by Eq. 10.2, each component possesses the same importance in the final result.

In a study describing the candidates to the 2004 US presidential election, Slatcher et al. [363] suggest a variable to indicate the *Presidentiality* of a candidate by the following equation:

$$\begin{aligned} Presidentiality = {} & Z\ score(Article) + Z\ score(Preposition) \\ & + Z\ score(Posemo) + Z\ score(BigWords) \end{aligned} \tag{10.3}$$

As shown in this chapter, this measurement will favor past presidents writing with more determiners (see Fig. 10.1), prepositions, and including a higher percentage of

big words (see Fig. 10.2). A higher frequency of positive emotional words can be however found in the last presidencies, mainly from Reagan to Trump as reported in Table 10.4.

Just to provide an example, this *Presidentiality* measure applied to all US presidents indicates in the first three ranks Madison (1809–1817), Hayes (1877–1891), and Adams (1797–1801). Usually nominated as the three best US presidents [318], Washington (1789–1797) appears at the 10th rank, Lincoln (1861–1865) in the 33rd, and Roosevelt (1933–1945) in the 36th. In the last two ranks, one can see H. Bush and Obama, both depicting a very low frequency of articles, prepositions, and big words together with a high percentage of positive emotional terms (around 1.4%). This last component cannot compensate for the three others. Clearly the concept of brilliant style is changing over time, and what was in fashion in the nineteenth century is no longer viewed as a bright writing style.

However, a word of caution is in order. Text analysis with a set of wordlists is not without concerns. First, one can emphasize that a term could have more than one meaning and this language ambivalence is ignored by a simple count of the number of occurrences. For example, the word *power* covers distinct meanings in the expressions *power plant* and *in power*. Second, the meaning attached to a word is evolving with time [76], and for some of them, the sense might diverge largely between Washington's and Trump's presidencies. Third, the short context of word is disregarded by a simple word count procedure. For example, the expressions *I'm happy* and *I'm not happy* are in opposition. The occurrence of the adjective *happy* is therefore not always an indication of a positive emotion. Taking account of multiword expressions is a partial solution to such drawbacks [419].

Fourth, the spelling must be strictly respected because small variations could represent different meanings (e.g., US or U.S. and us (as a pronoun)). Fifth, all languages include idiomatic constructions that must not be literality interpreted such as in *like a bull in a china-shop* or *tears of joy*. In this last example, one must not count this expression for one negative (*tears*) and one positive sentiment (*joy*). Sixth, the establishment of such wordlists is not without difficulties. It is not always clear whether a word expressed the specified emotion. For example, the words *die* or *bury* should appear in a list about death, but including other forms could be questionable (e.g., *grief*, *expire*, or *pass on*). As another example, the adverb *unbelievable* occurring frequently in Trump's tweets is not included in any list. Could one assume that it is marginal and does not affect the conclusions that can be drawn, or one needs to include it?

Finally, another questionable assumption is the additive hypothesis. Each occurrence of a word increases the category intensity by the same amount. In all languages, some words have a greater valence than others (e.g., incident, accident, or disaster). Moreover, the marginal increase in intensity may be more important for the first occurrences than for a word always having a high frequency. These aspects are not directly taken into consideration.

These remarks about the limits of the proposed approach should not invalidate all the results that can be drawn. Both LIWC and Diction systems have been used in many studies and can provide useful insights on stylistic aspects. And the choice

of the terms and vocabulary is of prime importance in politics. For example, if the world is talking about covid-19 (the illness) caused by the coronavirus (because its form is similar to a crown), Trump's administration imposes the use of "Wuhan virus" [174] to insist on the origin of the disease.

## 10.6 Conclusion

As described in this chapter, the application of various techniques and methods related to stylometry can be performed in various domains and topics. As shown, even simple tools such as the evolution of the relative frequency can provide useful information about the stylistic trends or term frequency evolution over the years. Moreover, one can illustrate an underlying tendency by considering two factors as depicted in Fig. 10.2. With PCA, one can exemplify the similarities when considering several stylistic and grammatical features. Of course, there is no guarantee that the resulting graph will take account of a large part of the underlying variability. With the increased ubiquity of the R software, new efficient tools can be made available producing better text processing and representations of the results (e.g., in the form of 2D graphs). More advanced stylometric models should be applied to reveal hidden patterns in the textual datasets as shown with the characteristic vocabulary and the identification of typical sentences in the last two sections. But we have just explained a few ways to apply those tools, and the reader is invited to find other applications and to propose new metrics and tools (see, for example, [405]). The stylometry, and in general the digital humanities, is a large open field still to be explored.

# Chapter 11
# Conclusion


As presented in the previous chapters, stylometric models and applications are located at the crossroads of several domains such as applied linguistics, statistics, and computer science. This position is not unique but, in a broader view, it corresponds to digital humanities, a field largely open to many relevant research directions and useful applications. These considerations lead to one of the main intents of this book: an introduction in this joint open discipline bringing together varied skills and requiring multi-disciplinary knowledge. Nowadays, we are just at the beginning of exploring all the potential of computer-based tools to represent, explore, understand, and identify patterns in literary textual datasets as well in other corpus formats.

In the near future, further research should improve existing stylometric models. When considering the authorship attribution question, for example, new research should be undertaken to identify other forms of stylistic markers or attribution schemes. Following [106] or [394], the vocabulary size known by an author could generate useful complementary evidence during an attribution process. As other improvements, new combinations of stylistic features could present more successful textual representation. In addition to discovering new stylistic indicators, one can focus on more effective matching strategies to identify with a higher probability the real author or some stylistic categories (e.g., Is this customer's review a fabrication or reflects a real experience?). New deep learning models and networks might prove effective to such research avenues but some enhancements can also be suggested for existing methods.

Second, we certainly need to achieve a better understanding of the effectiveness of existing text representation strategies as well as distance or similarity computations in numerous contexts. Published studies mainly focus on literary works (e.g., Shakespeare's plays [104, 239], Restoration poems [46], Italian novels [406], as well as novel excerpts written in English or French [211]). Many other textual dimensions must be explored and analyzed before being able to guarantee a foolproof checklist. This study also includes effective accuracy rate estimations



J. Savoy, *Machine Learning Methods for Stylometry*,
https://doi.org/10.1007/978-3-030-53360-1_11

for real forensic applications [54]. Experiments must be conducted on various text genres [331] (e.g., social network communication channels, newspapers, e-mails, letters, and even oral transcriptions). Moreover, other natural languages must be studied, and particularly those that do not belong to the Indo-European family (e.g., Arabic, Finnish, Chinese, Japanese, Korean, etc.). The time factor could also reveal pertinent linguistic changes, for example, the literary evolution exposed in [328].

Third, even though current stylometric models have been designed and evaluated with specific objectives, they can be applied in other situations. Digital humanities can be explored in various directions using statistical and computer-based models. For example, instead of being limited to text, one can apply similar feature extraction strategies and distance measurements for identifying the composer of a piece of music [18, 23, 131] or when considering paintings [178, 277].

Fourth, author profiling studies indicate that some demographics about the writer can be identified, such as his gender, age range, psychological traits, social origin, native language, etc. [280, 299]. These findings, even imperfect, could be enlarged to explore new applications, including to learn whether Juliet's parlance (in the play *Romeo and Juliet*) closely reflects a feminine figure. As the mental state of an author can be partially detected in his writings [299], an early detection of signs of depression (and suicidal risk and ideation [85]) as well as of eating disorders (e.g., anorexia, bulimia) [143] is essential for allowing a timely diagnosis and an efficient therapy. In this perspective, a series of CLEF evaluation campaigns have been conducted [244, 245] proposing useful test corpora, evaluation metrics, and some preliminary findings. However, many questions and issues are still open.

Fifth, stylistic choices are not limited to only aesthetic justification. A given style can generate a wished-for tone and support rhetoric to convince an audience. During an electoral campaign or while in power, this objective is of prime interest for politicians [154, 363]. As an interesting example, the term "global warming" can be substituted by "climate change" [232]. The second term no longer indicates a clear direction, opening the possibility of a possible minor climate change or even a decrease in the temperature. Another research direction is studying the different ways used to build an argumentation or justification, for example, in an encyclopedia, comparing the stylistic forms used in the nineteenth century to those appearing in Wikipedia. In this perspective, one can investigate how the author indicates his certainty or doubt, or how the justifications are presented, etc. For another example, investigations could reveal the stylistic markers associated with hoaxes [4] as well as with disinformation (e.g., in newspapers, and especially on social networks and micro-blogging) [87, 129, 359, 360, 397]. To achieve this objective, the first focus would be on detecting hate speech [62] or whether or not a text (or set of tweets) has been generated by a computer (see Chap. 9). As demonstrated in [231], an effective approach to detect scientific papers generated by machine is also effective to identify incorrect nucleotide sequences.

Sixth, research must be undertaken to provide better comprehensible justification for any attribution proposed by computers. A useful text categorization model cannot be a simple black box producing an assignment without further explanation. Interpretability is essential for a technology to be adopted by human beings. Such

a justification must be provided in plain English, related to a linguistic theory with references to high-level stylistic variables. Such a stylometric theory must be able to provide an explanation about the observed stylistic features and their relationships. For example, when an author opts more frequently for the pattern "noun *of the* noun" instead of "noun noun," a larger number of prepositions and definite determiners are related to this syntactic choice. Another example, Chap. 8 indicates that Elena Ferrante is writing more often about *padre* (father) and *madre* (mother) and this finding corresponds to a specific Ferrante (and Starnone) stylistic marker. In following this perspective, Juola [194] suggests considering the frequency occurrence difference of groups of synonyms[1] as a way to define higher linguistic variables. Other high-level variables can be defined using wordlists generated manually (such as those presented in Chap. 10).

As a final remark, it is essential to ground a discovery on several bodies of evidence. Stylometric history presents a case in which an authorship attribution method called CUSUM was suggested as very effective. Additional experiments have been performed and conducted to revise the effectiveness of this model [51, 149, 166]. Thus *proof by experiment* should be undertaken with all the required rigor and applied to more than one corpus. Indeed, some unknown characteristics of a test collection as well as a given text genre (or communication channel, etc.) could favor a new attribution strategy (or a new feature set or a new feature selection approach) over the others. As shown in [365] and [366], other pitfalls can be encountered in a scientific investigation.

There are literally innumerable applications for these powerful and evolving new tools, limited only by researchers' imaginations and ingenuity. We hope this book stimulates the reader to have a better understanding of the main principles and applications of text categorization models based on stylistic features.

[1]Such groups could be defined by using the WordNet thesaurus [114] and its *synsets* or, for other European languages, the EuroWordNet thesaurus [417].

# Appendix A

## A.1 Additional Resources and References

There are several useful books presenting a good introduction to R and the basics of statistics in the context of linguistics data. We mention the following ones:

- Jockers, M. L. (2013). *Macroanalysis: Digital Methods and Literary History*. University of Illinois Press, Urbana.
- Jockers, M. L. (2014). *Text Analysis with R for Students of Literature*. Springer, Heidelberg.
- Desagulier, G. (2017). *Corpus Linguistics and Statistics with R*. Springer, Heidelberg.
- Gries, S. T. (2017). *Quantitative Corpus Linguistics with R: A Practical Introduction*. 2nd edition, Routledge, New York.
- Braun, W. J. & Murdoch, D. J. (2007). *A First Course in Statistical Programming with R*. Cambridge University Press, Cambridge. (In this case, the focus is only on R and statistics.)

For an essential source of information about stylometry, stylistic studies, and its applications, see articles published in the following scientific journals:

- *Digital Scholarship in the Humanities* (DSH), Oxford University Press, UK (previously entitled *Literary and Linguistic Computing*).
- *Journal of Quantitative Linguistics*, Routledge, UK.
- *Language Resources and Evaluation*, Springer, Germany (previously, this journal was called *Computers and the Humanities*).
- *Journal of the Association for Information Science and Technology*, John Wiley, USA.
- *Information Processing & Management*, Elsevier, Netherlands.
- *Computational Linguistics Journal*, ACL, USA.
- *Plos ONE*, PLOS, USA.



J. Savoy, *Machine Learning Methods for Stylometry*,
https://doi.org/10.1007/978-3-030-53360-1

- *PNAS*, Proceedings of the National Academy of Sciences of the United States of America, USA.
- *Glottometrics*, RAM-Verlag, USA.
- *English Studies*, Taylor & Francis, UK.
- *Quality & Quantity*, Springer, Germany.
- *Frontiers in Digital Humanities*, Frontiers Media, Switzerland.
- At www.zotero.org/groups/643516/stylometry_bibliography, one can find a comprehensive bibliography on stylometry.

Of international conferences, the annual CLEF (Conference and Labs of the Evaluation Forum) is notable, within which the PAN track is dedicated to stylometry applications (plagiarism detection, authorship identification, author profiling, author verification, author clustering, obfuscation, etc.). The CLEF PAN creates each year new test collections written in different Indo-European languages. This conference also produces notebooks (denoted Working Papers) describing the datasets, presenting an overview of each track and sub-track. Moreover, the solution proposed by each participating team is provided in a separate article. A second published proceedings appears in the series LNCS by Springer with peer-reviewed papers.

One relevant association, the IQLA (International Quantitative Linguistics Association), is a European group of scientists in stylometry that organizes each two years an International Quantitative Linguistics Conference (QUALICO).

The ACL (Association for Computational Linguistics) publishes the *Computational Linguistics Journal* and several international annual conferences on NLP topics, such as EMNLP (Empirical Methods in Natural Languages Processing), NAACL (North American Chapter of the ACL), or CoNLL (Conference on Computational Natural Language Learning).

Another international conference, the annual DH (Digital Humanities) conference, offers a focus on humanities in general. The proceedings, usually online, publish only extended abstracts of the selected presentations. In addition to the conference, several SIGs (Special Interest Groups) organize workshops on more specific topics.

Some papers related to stylometry and text categorization appear in the European Conference on Information Retrieval (ECIR) and sometimes in the ACM-SIGIR conference (Special Interest Group in Information Retrieval).

## A.2 The Most Frequent Word-Types in the *Federalist Papers*

As the *Federalist Papers* corresponds to our running example, more information about its contents is provided in this appendix. First, the top 50 most frequent terms are provided with their occurrence frequency in Table A.1. As one can see, the large majority of them are functional terms (determiners, prepositions, conjunctions, personal pronouns, auxiliary verb forms, or modal verbs) as well as punctuation symbols. This list as well as other lists in this appendix is based on the articles authored by Hamilton, Madison, and Jay, but without the article titles, the pen name

"Publius," and the sentence "To the People of the State of New York:" appears in all articles.

**Table A.1** The 50 most frequent terms in the *Federalist Papers*

| Freq. | Token | Freq. | Token | Freq. | Token | Freq. | Token | Freq. | Token |
|---|---|---|---|---|---|---|---|---|---|
| 14,578 | the | 2158 | it | 1023 | for | 713 | on | 522 | has |
| 10,963 | , | 1834 | is | 1012 | not | 687 | government | 495 | if |
| 9771 | of | 1704 | which | 1010 | will | 687 | been | 475 | at |
| 6013 | to | 1425 | as | 890 | their | 675 | may | 471 | more |
| 4261 | . | 1379 | by | 879 | with | 604 | all | 465 | any |
| 4242 | and | 1228 | ; | 860 | from | 598 | state | 463 | than |
| 3745 | in | 1187 | this | 830 | are | 578 | but | 455 | them |
| 3340 | a | 1148 | would | 811 | an | 558 | its | 447 | one |
| 3184 | be | 1088 | have | 794 | they | 553 | power | 429 | those |
| 2371 | that | 1064 | or | 713 | states | 524 | other | 427 | no |

In Table A.2, the 20 most frequent terms per author are reported with their occurrence frequency. Clearly, these lists are very similar between Hamilton and Madison but differ slightly with Jay's list.

**Table A.2** The 20 most frequent terms in the *Federalist Papers* per author

| | Hamilton | | Madison | | Jay | | Disputed | |
|---|---|---|---|---|---|---|---|---|
| Rank | Freq. | Token | Freq. | Token | Freq. | Token | Freq. | Token |
| 1 | 10,186 | the | 3876 | the | 635 | , | 2271 | the |
| 2 | 7504 | , | 2824 | , | 516 | the | 1681 | , |
| 3 | 7106 | of | 2306 | of | 408 | and | 1421 | of |
| 4 | 4478 | to | 1247 | to | 359 | of | 754 | to |
| 5 | 2996 | . | 1163 | and | 288 | to | 710 | . |
| 6 | 2773 | in | 1066 | . | 199 | . | 591 | and |
| 7 | 2671 | and | 808 | in | 164 | in | 583 | be |
| 8 | 2472 | a | 768 | a | 160 | be | 531 | in |
| 9 | 2270 | be | 754 | be | 150 | that | 528 | a |
| 10 | 1679 | that | 542 | that | 138 | it | 340 | that |
| 11 | 1523 | it | 497 | it | 102 | as | 309 | it |
| 12 | 1296 | is | 481 | is | 100 | a | 286 | is |
| 13 | 1224 | which | 452 | by | 90 | or | 286 | which |
| 14 | 951 | as | 424 | which | 87 | they | 278 | by |
| 15 | 921 | would | 379 | ; | 83 | their | 241 | as |
| 16 | 900 | this | 372 | as | 82 | by | 227 | will |
| 17 | 845 | by | 295 | on | 73 | will | 188 | on |
| 18 | 796 | ; | 265 | have | 68 | would | 177 | not |
| 19 | 770 | have | 259 | for | 65 | not | 172 | for |
| 20 | 742 | or | 256 | not | 60 | with | 163 | this |

In Table A.3, the 20 most frequent bigrams of letters in the overall corpus (with the 12 disputed articles) and per author are reported with their occurrence frequency. In these sequences of letters, the symbol _ indicates the space between tokens, used to delimit each word-type. For all three authors, the most frequent bigram is the letter "e" at the end of a word and the second most frequent is the letter "t" in the beginning of a word. When composed of two letters, the most frequent bigram is "th" (which is not a surprise due to the very frequent occurrences of the determiners *the*, *that*, or *this*).

**Table A.3** The 20 most frequent bigrams of letters in the *Federalist Papers* per author

| | Corpus | | Hamilton | | Madison | | Jay | |
|---|---|---|---|---|---|---|---|---|
| Rank | Freq. | Bigram | Freq. | Bigram | Freq. | Bigram | Freq. | Bigram |
| 1 | 41,790 | e_ | 25,512 | e_ | 8841 | e_ | 1727 | e_ |
| 2 | 33,498 | _t | 20,766 | _t | 7026 | _t | 1438 | _t |
| 3 | 28,336 | th | 17,077 | th | 6272 | th | 1227 | th |
| 4 | 23,582 | he | 14,068 | he | 5341 | he | 1030 | _a |
| 5 | 22,047 | s_ | 13,157 | s_ | 4922 | s_ | 1024 | s_ |
| 6 | 19,789 | _a | 11,970 | _a | 4229 | _a | 980 | he |
| 7 | 18,169 | t_ | 11,032 | t_ | 3759 | t_ | 932 | d_ |
| 8 | 17,761 | _o | 10,948 | n_ | 3756 | _o | 929 | t_ |
| 9 | 17,219 | n_ | 10,910 | _o | 3679 | er | 756 | _o |
| 10 | 15,437 | er | 9197 | on | 3490 | n_ | 716 | er |
| 11 | 15,343 | d_ | 9079 | d_ | 3425 | d_ | 707 | an |
| 12 | 14,884 | on | 8907 | er | 3170 | on | 683 | n_ |
| 13 | 14,049 | in | 8720 | in | 2975 | in | 675 | on |
| 14 | 13,172 | re | 8306 | _i | 2806 | ,_ | 653 | re |
| 15 | 13,125 | _i | 7720 | re | 2797 | re | 649 | in |
| 16 | 12,596 | ,_ | 7660 | f_ | 2659 | _i | 632 | ,_ |
| 17 | 12,445 | at | 7604 | at | 2659 | ti | 611 | nd |
| 18 | 12,086 | y_ | 7480 | ,_ | 2650 | y_ | 607 | y_ |
| 19 | 12,030 | f_ | 7369 | of | 2573 | at | 602 | at |
| 20 | 12,026 | ti | 7330 | ti | 2516 | an | 596 | en |

In Table A.4, the 20 most frequent trigrams of letters in the overall corpus (with the 12 disputed articles) and per author are shown with their occurrence frequency. On the top of those lists, one can see the importance of the determiners *the*, *that*, or *this* in the English language. As the most frequent trigrams composed only of letters, one can see "ion," "and," "tio," or "ent."

**Table A.4** The 20 most frequent trigrams of letters in the *Federalist Papers* per author

| | Corpus | | Hamilton | | Madison | | Jay | |
|---|---|---|---|---|---|---|---|---|
| Rank | Freq. | Trigram | Freq. | Trigram | Freq. | Trigram | Freq. | Trigram |
| 1 | 24,427 | _th | 14,814 | _th | 5361 | _th | 1026 | _th |
| 2 | 20,557 | the | 12,267 | the | 4668 | the | 836 | the |
| 3 | 16,918 | he_ | 10,258 | he_ | 3838 | he_ | 627 | _,_ |
| 4 | 12,552 | _,_ | 7451 | _,_ | 2798 | _,_ | 521 | he_ |
| 5 | 11,468 | _of | 7309 | _of | 2352 | _of | 470 | nd_ |
| 6 | 11,141 | of_ | 7077 | of_ | 2292 | of_ | 452 | _an |
| 7 | 7444 | ion | 4798 | ion | 1476 | ion | 431 | and |
| 8 | 6955 | _in | 4583 | _to | 1475 | ed_ | 383 | _of |
| 9 | 6940 | to_ | 4550 | to_ | 1462 | _an | 357 | of_ |
| 10 | 6936 | _to | 4454 | _in | 1452 | on_ | 327 | _in |
| 11 | 6872 | on_ | 4378 | on_ | 1358 | nd_ | 305 | to_ |
| 12 | 6474 | _an | 3864 | tio | 1334 | _in | 299 | ion |
| 13 | 6124 | ed_ | 3758 | _an | 1305 | to_ | 293 | _to |
| 14 | 6070 | tio | 3512 | ed_ | 1289 | _to | 277 | ed_ |
| 15 | 5888 | nd_ | 3379 | _be | 1247 | and | 261 | es_ |
| 16 | 5542 | _be | 3370 | f_t | 1224 | tio | 256 | _co |
| 17 | 5257 | f_t | 3343 | nd_ | 1172 | ent | 239 | ent |
| 18 | 5240 | and | 3215 | _co | 1153 | es_ | 238 | tio |
| 19 | 5114 | es_ | 2963 | es_ | 1148 | er_ | 238 | on_ |
| 20 | 5085 | _co | 2957 | in_ | 1131 | _be | 224 | _be |

## A.3 Proposed Features for the *Federalist Papers*

Various studies have proposed a reduced set of features to solve the *Federalist Papers* question. The shortest list was proposed by Bosch and Smith [38] with three words (namely *are*, *our*, and *upon*) and a similar number is suggested by Fung [124] (*to*, *upon*, and *would*). Matthews and Merriam's [254] also explained how only five terms are useful to discriminate between Hamilton and Madison (*are*, *in*, *no*, *of*, and *the*). Holmes and Forsyth [162] proposed a list of eight words (*both*, *by*, *consequently*, *kind*, *on*, *there*, *upon*, and *whilst*) while Tweetie et al. [410] proposed eleven words (*an*, *any*, *can*, *do*, *every*, *from*, *his*, *may*, *on*, *there*, and *upon*). Finally, Kacmarcik and Gamon [196] have used fourteen terms to discriminate between Hamilton and Madison (*and*, *any*, *at*, *by*, *:*, *in*, *less*, *men*, *on*, *powers*, *there*, *those*, *to*, and *upon*). All these examples based on a reduced feature set must be taken with caution. Limited to only a few words, the classifier is certainly overfitted for this dataset. In addition, one can easily write a text imitating Madison or Hamilton's style when the authorship attribution system is focusing only on a few terms.

The most well-known study on feature selection for the *Federalist Papers* was conducted by Mosteller and Wallace [273], who first suggested a long list of 70 words (*a*, *all*, *also*, *an*, *and*, *any*, *are*, *as*, *at*, *be*, *been*, *but*, *by*, *can*, *do*, *down*, *even*,

*every*, *for*, *from*, *had*, *has*, *have*, *her*, *his*, *if*, *in*, *into*, *is*, *it*, *its*, *may*, *more*, *must*, *my*, *no*, *not*, *now*, *of*, *on*, *only*, *or*, *our*, *shall*, *should*, *so*, *some*, *such*, *than*, *that*, *the*, *their*, *then*, *there*, *things*, *this*, *to*, *up*, *upon*, *was*, *were*, *what*, *when*, *which*, *who*, *will*, *with*, *would*, and *your*).

A shorter one, called the final list, contains 35 terms (*upon*, *also*, *an*, *by*, *of*, *on*, *there*, *this*, *to*, *although*, *both*, *enough*, *while*, *whilst*, *always*, *though*, *commonly*, *consequently*, *considerable(ly)*, *according*, *apt*, *direction*, *innovation(s)*, *language*, *vigor(ous)*, *kind*, *matter(s)*, *particularly*, *probability*, and *work(s)*). As one can see, not all of them can be considered as functional terms (e.g., *language*).

## A.4 Feature Selection

This section presents the top 10 most discriminative word-types according to the two possible authors, namely Hamilton and Madison. For each author, six feature-scoring functions have been applied, namely the chi-square (CHI), the information gain (IG), the gain ratio (GR), the pointwise mutual information (PMI), the odds ratio (OR), and the GSS coefficient (see Tables A.5 and A.6 for Hamilton, and, for Madison, the Tables A.7 and A.8.).

**Table A.5** The top 10 most discriminative word-types for Hamilton

| Chi-square | | | Information Gain | | | Gain Ratio | | |
|---|---|---|---|---|---|---|---|---|
| Score | Freq. | Term | Score | Freq. | Term | Score | Freq. | Term |
| 47.0 | 370/9 | upon | 0.50 | 370/9 | upon | 0.23 | 370/9 | upon |
| 30.5 | 1/11 | although | 0.29 | 1/11 | although | 0.13 | 1/11 | although |
| 23.4 | 4/18 | consequently | 0.22 | 4/18 | consequently | 0.09 | 4/8 | wish |
| 23.1 | 4/8 | wish | 0.22 | 4/8 | wish | 0.09 | 4/18 | consequently |
| 19.9 | 1/19 | whilst | 0.19 | 1/19 | whilst | 0.08 | 1/19 | whilst |
| 17.4 | 6/22 | absolutely | 0.17 | 24/0 | readily | 0.08 | 0/22 | assumed |
| 16.6 | 1/18 | composing | 0.16 | 52/46 | many | 0.08 | 0/9 | enlarge |
| 16.6 | 1/9 | recommended | 0.16 | 6/22 | absolutely | 0.08 | 0/5 | universally |
| 15.0 | 12/23 | formed | 0.15 | 1/18 | composing | 0.08 | 0/2 | gentlemen |
| 14.9 | 14/30 | paper | 0.15 | 1/9 | recommended | 0.08 | 0/7 | indispensably |

In the following tables, each cell indicates first the score achieved by the corresponding word-types with respect to the local utility function, and then the occurrence frequency of this term is provided in both articles written by Hamilton and by Madison. This information gives an indication of the term importance in the corpus and its distribution between the two possible authors. For example, in

**Table A.6** The top 10 most discriminative word-types for Hamilton (cont.)

| PMI | | | Odds Ratio | | | GSS | | |
|---|---|---|---|---|---|---|---|---|
| Score | Freq. | Term | Score | Freq. | Term | Score | Freq. | Term |
| 0.46 | 1/0 | criticism | 17.9 | 378/54 | there | 0.15 | 370/9 | upon |
| 0.46 | 2/0 | golden | 16.0 | 35/0 | intended | 0.09 | 76/1 | kind |
| 0.46 | 1/0 | charter | 9.8 | 22/1 | mentioned | 0.08 | 35/0 | intended |
| 0.46 | 4/0 | destined | 9.8 | 22/0 | commonly | 0.08 | 69/3 | community |
| 0.46 | 4/0 | revolt | 9.0 | 19/1 | about | 0.08 | 24/0 | readily |
| 0.46 | 1/0 | preface | 8.2 | 76/1 | kind | 0.08 | 102/21 | man |
| 0.46 | 1/0 | pulse | 8.2 | 46/2 | matter | 0.08 | 46/2 | matter |
| 0.46 | 1/0 | patrimony | 7.6 | 32/3 | apt | 0.07 | 32/3 | apt |
| 0.46 | 3/0 | mistakes | 7.5 | 20/2 | forward | 0.07 | 47/8 | considerable |
| 0.46 | 2/0 | hanging | 6.3 | 69/3 | community | 0.07 | 33/0 | enough |

Table A.5 under the column "Chi-square" one can find the first cell related to the word-type *upon*. This term achieved a local score of 47.0 and appears 370 times in papers authored by Hamilton and only 9 times in Madison's articles.

The intersection is clearly larger between the top 10 most discriminative terms retrieved by the Chi-square, IG, and GR functions compared to the three others. These latter functions tend to promote distinct terms and the intersection between terms selected by each function is rather small.

**Table A.7** The top 10 most discriminative word-types for Madison

| Chi-square | | | Information Gain | | | Gain Ratio | | |
|---|---|---|---|---|---|---|---|---|
| Score | Freq. | Term | Score | Freq. | Term | Score | Freq. | Term |
| 30.6 | 1/19 | whilst | 0.25 | 1/19 | whilst | 0.18 | 36/35 | few |
| 28.9 | 370/9 | upon | 0.25 | 370/9 | upon | 0.18 | 370/9 | upon |
| 25.7 | 1/18 | composing | 0.22 | 36/35 | few | 0.16 | 1/19 | whilst |
| 21.5 | 0/12 | assumed | 0.21 | 1/18 | composing | 0.16 | 76/1 | kind |
| 21.5 | 0/9 | enlarge | 0.19 | 76/1 | kind | 0.13 | 6/22 | absolutely |
| 21.5 | 0/7 | indispensably | 0.19 | 6/22 | absolutely | 0.13 | 6/20 | proceedings |
| 21.5 | 0/6 | administering | 0.19 | 6/20 | proceedings | 0.13 | 4/18 | consequently |
| 21.5 | 0/7 | violating | 0.19 | 4/18 | consequently | 0.13 | 1/18 | composing |
| 21.5 | 6/22 | absolutely | 0.18 | 0/12 | assumed | 0.13 | 35/0 | intended |
| 21.5 | 6/20 | proceedings | 0.18 | 0/9 | enlarge | 0.12 | 10/24 | particularly |

**Table A.8** The top 10 most discriminative word-types for Madison (cont.)

| PMI | | | Odds Ratio | | | GSS | | |
|---|---|---|---|---|---|---|---|---|
| Score | Freq. | Term | Score | Freq. | Term | Score | Freq. | Term |
| 2.3 | 0/1 | misconstruction | 73.3 | 1/19 | whilst | 0.09 | 36/35 | few |
| 2.3 | 0/2 | disfigured | 55.0 | 1/18 | composing | 0.09 | 10/24 | particularly |
| 2.3 | 0/2 | rewards | 41.3 | 1/13 | sphere | 0.09 | 6/22 | absolutely |
| 2.3 | 0/2 | echoed | 30.6 | 1/7 | viewed | 0.09 | 1/19 | whilst |
| 2.3 | 0/1 | equalized | 30.6 | 1/7 | pronounced | 0.09 | 6/20 | proceedings |
| 2.3 | 0/2 | animate | 30.6 | 1/8 | relief | 0.09 | 4/18 | consequently |
| 2.3 | 0/4 | shoots | 30.6 | 1/10 | respectively | 0.09 | 13/32 | fully |
| 2.3 | 0/2 | toleration | 30.6 | 1/6 | planned | 0.08 | 14/30 | paper |
| 2.3 | 0/1 | venial | 22.0 | 1/5 | ten | 0.08 | 1/18 | composing |
| 2.3 | 0/1 | spectacles | 22.0 | 1/5 | pieces | 0.08 | 17/24 | regular |

## A.5 Most Frequent Terms in Italian

Based on the Italian corpus used in Chap. 8, one can derive the 50 most frequent word-tokens as shown in Table A.9. In this list, one can find several definite determiners (e.g., *la*, *lo*, *il*, *le*, *i*, *l*, *gli* (the)), indefinite determiners (e.g., *un*, *una* (a/an)), some pronouns (e.g., *io* (I), *lei* (you), *mi* (me), *suo* (his), *sua* (her)), as well as some prepositions (e.g., *di* (of), *a* (to), *da* (from), *con* (with), *per* (for), *come* (as), *solo* (only)), a few adverbs (e.g., *si* (yes), *non* (not)), the most frequent forms related to the verb *to be* (e.g., *sono* (I am), *è* (s/he is), *era* (was)), or *to have* (e.g., *ha* (s/he has), *ho* (I have), *aveva* (had)), and finally some conjunctions (e.g., *che* (that), *ma* (but), *e* (and), *o* (or)).

**Table A.9** The 50 most frequent terms in our Italian corpus

| Freq. | Token | Freq. | Token | Freq. | Token | Freq. | Token | Freq. | Token |
|---|---|---|---|---|---|---|---|---|---|
| 312,846 | di | 102,714 | si | 57,405 | ma | 36,957 | ha | 25,214 | lui |
| 280,539 | e | 102,076 | una | 55,178 | come | 32,709 | sono | 25,196 | lei |
| 244,771 | che | 98,236 | le | 53,487 | del | 31,005 | ci | 24,549 | c |
| 198,211 | la | 92,610 | è | 50,785 | lo | 29,070 | io | 24,392 | nel |
| 186,661 | a | 85,080 | era | 48,398 | se | 29,036 | ho | 24,215 | o |
| 171,127 | il | 83,037 | con | 48,153 | gli | 28,734 | alla | 23,477 | sua |
| 166,479 | un | 81,822 | l | 45,108 | della | 28,658 | anche | 23,239 | dei |
| 158,214 | non | 77,693 | mi | 44,151 | aveva | 27,583 | quando | 22,712 | suo |
| 127,485 | in | 65,510 | i | 44,107 | più | 26,081 | poi | 21,331 | così |
| 103,351 | per | 60,173 | da | 40,197 | al | 26,023 | perché | 21,298 | solo |

## A.6 US Presidents

Table A.10 depicts background information about the US presidents. When looking at the US political parties, we encounter first the Federalist (F), which will disappear around 1812. G. Washington seats as independent (Ind.) but was close to the Federalists' ideas. Its rival was the Democratic–Republican (D–R) Party that will split in two in 1825 to form the Democrat Party (D) and the National Republican (N–R). This latter party, dissolved in 1833, will be followed by the Whig Party. In 1854, members of the Whig Party founded the Republican (R) Party, which takes the lead over the Whig movement.

**Table A.10** List of 45 US Presidents with their number of Inaugural and SOTU speeches together with their political affiliation

| No. | Name | Inaugural | SOTU | From | To | Party |
|---|---|---|---|---|---|---|
| 1 | George Washington | 2 | 8 | 1789 | 1797 | Ind. |
| 2 | John Adams | 1 | 4 | 1797 | 1801 | F |
| 3 | Thomas Jefferson | 2 | 8 | 1801 | 1809 | D-R |
| 4 | James Madison | 2 | 8 | 1809 | 1817 | D-R |
| 5 | James Monroe | 2 | 8 | 1817 | 1825 | D-R |
| 6 | John Quincy Adams | 1 | 4 | 1825 | 1829 | N-R |
| 7 | Andrew Jackson | 2 | 8 | 1829 | 1837 | D |
| 8 | Martin Van Buren | 1 | 4 | 1837 | 1841 | D |
| 9 | William H. Harrison | 1 | | 1841 | 1841 | Whig |
| 10 | John Tyler | | 4 | 1841 | 1845 | D |
| 11 | James Polk | 1 | 4 | 1845 | 1849 | D |
| 12 | Zachary Taylor | 1 | 1 | 1849 | 1850 | Whig |
| 13 | Millard Fillmore | | 3 | 1850 | 1853 | Whig |
| 14 | Franklin Pierce | 1 | 4 | 1853 | 1857 | D |
| 15 | James Buchanan | 1 | 4 | 1857 | 1861 | D |
| 16 | Abraham Lincoln | 2 | 4 | 1861 | 1865 | R |
| 17 | Andrew Johnson | | 4 | 1865 | 1869 | D |
| 18 | Ulysses S. Grant | 2 | 8 | 1869 | 1877 | R |
| 19 | Rutherford B. Hayes | 1 | 4 | 1877 | 1881 | R |
| 20 | James A. Garfield | 1 | | 1881 | 1881 | R |
| 21 | Chester A. Arthur | | 4 | 1881 | 1885 | R |
| 22 | Grover Cleveland | 1 | 4 | 1885 | 1889 | D |
| 23 | Benjamin Harrison | 1 | 4 | 1889 | 1893 | R |
| 24 | Grover Cleveland | 1 | 4 | 1893 | 1897 | D |
| 25 | William McKinley | 2 | 4 | 1897 | 1901 | R |
| 26 | Theodore Roosevelt | 1 | 8 | 1901 | 1909 | R |
| 27 | William H. Taft | 1 | 4 | 1909 | 1913 | R |
| 28 | Woodrow Wilson | 2 | 8 | 1913 | 1921 | D |

(continued)

**Table A.10** (continued)

| No. | Name | Inaugural | SOTU | From | To | Party |
|---|---|---|---|---|---|---|
| 29 | Warren Harding | 1 | 2 | 1921 | 1923 | R |
| 30 | Calvin Coolidge | 1 | 6 | 1923 | 1929 | R |
| 31 | Herbert Hoover | 1 | 4 | 1929 | 1933 | R |
| 32 | Franklin D. Roosevelt | 4 | 12 | 1933 | 1945 | D |
| 33 | Harry S. Truman | 1 | 8 | 1945 | 1953 | D |
| 34 | Dwight D. Eisenhower | 2 | 9 | 1953 | 1961 | R |
| 35 | John F. Kennedy | 1 | 3 | 1961 | 1963 | D |
| 36 | Lyndon B. Johnson | 1 | 6 | 1963 | 1969 | D |
| 37 | Richard Nixon | 2 | 5 | 1969 | 1974 | R |
| 38 | Gerald R. Ford | | 3 | 1974 | 1977 | R |
| 39 | Jimmy Carter | 1 | 4 | 1977 | 1981 | D |
| 40 | Ronald Reagan | 2 | 7 | 1981 | 1989 | R |
| 41 | George H. Bush | 1 | 4 | 1989 | 1993 | R |
| 42 | Bill Clinton | 2 | 8 | 1993 | 2001 | D |
| 43 | George W. Bush | 2 | 8 | 2001 | 2009 | R |
| 44 | Barack Obama | 2 | 8 | 2009 | 2017 | D |
| 45 | Donald Trump | 1 | 4 | 2017 | 2020 | R |

# References


1. A. Abbasi, H. Chen, Writeprints: a stylometric approach to identity-level identification and similarity detection in cyberspace. ACM Trans. Inf. Syst. **26**(2) (2008). Article 7
2. S. Adamovic, V. Miskovic, M. Milosavljevic, M. Sarac, M. Veinovic, Automated language-independent authorship verification (for Indo-European languages). J. Assoc. Inf. Sci. Technol. **70**(8), 858–871 (2019)
3. D. Adger, *Language Unlimited. The Science Behind Our Most Creative Power* (Oxford University Press, Oxford, 2019)
4. S. Afroz, M. Brennam, R. Greenstadt, Detecting hoaxes, frauds, and deception in writing style online, in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, pp. 402–416 (IEEE Computer Society, Washington, 2012)
5. C.C. Aggarwal, Mining text streams, in *Mining Text Data*, ed. by C.C. Aggarwal, C.X. Zhai (Springer, New York, 2012), pp. 297–321
6. S. Ahmadian, S. Azarshahi, D.L. Paulhus, Explaining Donald Trump via communication style: grandiosity, informality, and dynamism. Personal. Individ. Differ. **107**, 49–53 (2017)
7. N. Akiva, M. Koppel, Identifying distinct components of a multi-author document, in *European Intelligent and Security Informatics Conference* (2012), pp. 205–209
8. M. Alfaro, The daily 202: Alexander Hamilton has been cast in a starring for impeachment's closing argument, in *Washington Post*, 143 (Dec. 17th) (2019)
9. M. Almishari, G. Tsudik, Exploring linkability of user reviews, in *Proceedings Computer Security ESORICS*. Lecture Notes in Computer Science, vol. 7459 (Springer, Berlin, 2012), pp. 307–324.
10. S.M. Alzahrani, N. Salim, A. Abraham, Understanding plagiarism linguistic patterns, textual features, and detection methods. IEEE Trans. Syst. Man Cybernet. Part C (Appl. Rev.) **42**(2), 133–149 (2012)
11. A. Antonia, C. Hugh, J. Elliott, Language chunking, data sparseness, and the value of a long marker list: explorations with word n-grams and authorial attribution. Lit. Linguis. Comput. **29**(2), 147–163 (2014)
12. S. Argamon, Interpreting Burrows' Delta: geometric and probabilistic foundations. Lit. Linguist. Comput. **23**(2), 131–147 (2008)
13. S. Argamon, M. Koppel, J.W. Pennebaker, J. Schler, Automatically profiling the author of an anonymous text. Commun. ACM **52**(2), 119–123 (2009)
14. H.R. Baayen, *Word Frequency Distributions* (Kluwer Academic Press, Dordrecht, 2001)
15. H.R. Baayen, *Analysis Linguistic Data: A Practical Introduction to Statistics Using R* (Cambridge University Press, Cambridge, 2008)




J. Savoy, *Machine Learning Methods for Stylometry*,
https://doi.org/10.1007/978-3-030-53360-1

16. H. Baayen, H. van Halteren, F.J. Tweedie, Outside the cave of shadows: using syntactic annotation to enhance authorship attribution. Lit. Linguis. Comput. **11**(3), 121–132 (1996)
17. A. Bacciu, M. La Morgia, A. Mei, E. Nerio Nemmi, V. Neri, J. Stefa, Bot and gender detection of Twitter accounts using distortion and LSA. Notebook for PAN at CLEF 2019, in *Working Notes Papers of the CLEF 2019 Evaluation Labs volume 2380 of CEUR Workshop* (CEUR, Aachen, 2019)
18. E. Backer, P. van Kranenburg, On musical stylometry - A pattern recognition approach. Patt. Recogn. Lett. **26**(3), 299–309 (2005)
19. N. Bagnall, *Newspaper Language* (Focal Press, Oxford, 1993)
20. D.W. Barowy, E.D. Berger, B. Zorn, ExceLint: automatically finding spreadsheet formula errors, in *Proceedings ACM Programming Language*, vol. 2 (2018). Article 148
21. M. Barrick, M.K. Mount, The big five personality dimensions and job performance: a meta-analysis. Person. Psychol. **44**(1), 1–26 (1991)
22. L. Bauer, P. Trudgill, *Language Myths* (Penguin Books, London, 1998)
23. A. Bellaachia, E. Jimenez, Exploring performance-based music attributes for stylometric analysis. World Acad. Sci. Eng. Technol. **3**(7), 1795–1797 (2009)
24. D. Benedetto, E. Caglioti, V. Loreto, Language trees and zipping. Phys. Rev. Lett. **88**(4), 048702 (2002)
25. Y. Bengio, Learning deep architectures for AI. Found. Trends Mach. Learn. **2**(1), 1–127 (2009)
26. Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003)
27. I. Bensalem, P. Rosso, S. Chikhi, One the use of character n-grams as the evidence of plagiarism. Lang. Resour. Eval. **53**(2), 1–34 (2019)
28. S. Benzel, A simple stylometry comparator: Nifty assignment. J. Comput. Sci. Coll. **31**(2), 283–284 (2015)
29. D. Biber, Representativeness in corpus design. Lit. Linguis. Comput. **8**(4), 243–257 (1993)
30. D. Biber, *Dimensions of the Register Variation*. (Cambridge University Press, Cambridge, 1995)
31. D. Biber, S. Conrad, *Register, Genre, and Style* (Cambridge University Press, Cambridge, 2009)
32. D. Biber, S. Conrad, G. Leech, *The Longman Student Grammar of Spoken and Written English* (Longman, London, 2002)
33. J.N.G. Binongo, Who wrote the 15th *Book of Oz*? An application of multivariate analysis to authorship attribution. Chance **16**(2), 9–17 (2003)
34. J.N.G. Binongo, M.W. Smith, The application of principal component analysis to stylometry. Lit. Linguis. Comput. **14**(4), 445–465 (1999)
35. D.M. Blei, Probabilistic topic models. Commun. ACM **55**(4), 77–84 (2003)
36. D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation. Mach. Learn. **3**(1), 993–1022 (2003)
37. T. Bolukbasi, K.-W. Chang, J. Zou, V. Saligrama, A. Kalai, Man is to computer programmer as woman is to homemaker? Debiasing word embeddings, in *Advanced in Neural Information Processing Systems 29 (NIPS 2016)*, vol. 30 (The IEEE Press, Washington, 2016), pp. 4356–4364
38. R.A. Bosch, J.A. Smith, Separating hyperplanes and the authorship on the *Federalist Papers*. Am. Math. Mon. **105**(7), 601–608 (1991)
39. B.E. Boser, E. Sackinger, J. Bromley, Y. Le Cun, L.D. Jackel, An analog neural network processor with programmable topology. J. Solid State Circ. **26**(12), 2017–2025 (1991)
40. R.L. Boyd, J.W. Pennebaker, Language-based personality: a new approach to personality in a digital world. Curr. Opin. Behav. Sci. **18**, 63–68 (2017)
41. W.J. Braun, D.J. Murdoch, *A First Course in Statistical Programming with R* (Cambridge University Press, Cambridge, 2007)

42. M. Brennam, S. Afroz, R. Greenstadt, Adversarial stylometry: circumventing authorship recognition to preserve privacy and anonymity. ACM Trans. Inf. Syst. Secur. **13**(3) (2011). Article 12
43. L.D. Brown, T.T. Cai, A. DasGupta, Interval estimation for a binomial proportion. Stat. Sci. **16**(2), 101–133 (2001)
44. J.D. Burger, J. Henderson, G. Kim, G. Zarrella, Discriminating gender on Twitter, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2011), pp. 1301–1309
45. J.F. Burrows, Not unless you ask nicely: the interpretative Nexus between analysis and information. Lit. Linguis. Comput. **7**(1), 91–109 (1992)
46. J.F. Burrows, Delta: a measure of stylistic difference and a guide to likely authorship. Lit. Linguis. Comput. **17**(3), 267–287 (2002)
47. J.F. Burrows, All the way through: testing for authorship in different frequency strata. Lit. Linguis. Comput. **22**(1), 27–47 (2007)
48. J.W. Caesar, G.E. Thurow, J. Tulis, J.M. Bessette, The rise of rhetorical presidency. Pres. Stud. Q. **11**(2), 158–171 (1981)
49. C. Cai, L. Li, D. Zeng, Behavior enhanced deep bot detection in social media, in *Proceedings IEEE International Conference on Intelligence and Security Informatics (ISI)* (2017), pp. 128–130
50. F. Can, J.M. Patton, Change of writing style with time. Comput. Humanit. **38**(1), 61–82 (2004)
51. D.V. Canter, An evaluation of the "CUSUM" stylistic analysis of confessions. Expert Evid. **3**(1), 93–99 (1992)
52. S.-H. Cha, Comprehensive survey on distance similarity measures between probability density functions. Int. J. Math. Models Methods Appl. Sci. **1**(4), 300–307 (2007)
53. E. Charniak, *Introduction to Deep Learning* (The MIT Press, Cambridge, 2018)
54. C. Chaski, Best practices and admissibility of forensic author identification. J. Law Policy **21**(2), 333–376 (2013)
55. L. Chen, H. Zhang, J.M. Jose, H. Yu, Y. Moshfeghi, P. Triantafillou, Topic detection and tracking on heterogeneous information. J. Intell. Inf. Syst. **51**(1), 115–137 (2018)
56. Z. Chu, S. Gianvecchio, H. Wang, S. Jajodia, Detecting automation of twitter accounts: are you a human, bot, or cyborg? IEEE Trans. Dependable Secure Comput. **9**(6), 811–824 (2003)
57. K.W. Church, P. Hanks, Word association norms, mutual information, and lexicography, in *Proceedings Association for Computational Linguistics (ACL)*, pp. 76–83 (The ACL Press, Stroudsburg, 1999)
58. R. Cilibrasi, P.M.B. Vitanyi, Clustering by compression. IEEE Trans. Inf. Theory **51**(4), 1523–1545 (2005)
59. K. Connolly, *Der Spiegel* says top journalist faked stories for years. The Guardian, Dec. 19th, 2018
60. W.J. Conover, *Practical Nonparametric Statistics* (Wiley, New York, 1980)
61. G. Coppersmith, M. Dredze, C. Harman, Quantifying mental health signals in Twitter, in *ACL Workshop on Computational Linguistics and Clinical Psychology* (The ACL Press, Stroudsburg, 2014), pp. 51–60
62. M. Corazza, S. Menini, E. Cabrio, S. Tonelli, S. Villata, A multilingual evaluation for online hate speech detection. Lit. Linguis. Comput. **20**(2) (2020). Article 10
63. M.A. Cortelazzo, P. Nadalutti, A. Tuzzi, Improving Labbé intertextual distance: Testing a revised version on a large corpus of Italian literature. J. Quant. Linguis. **20**(2), 125–152 (2013)
64. M. Coulthard, On admissible linguistics evidence. J. Law Policy **21**(2) (2012). Article 8
65. H. Craig, A.F. Kinney, *Shakespeare, Computers, and the Mystery of Authorship* (Cambridge University Press, Cambridge, 2009)
66. M.J. Crawley, *Statistics. An Introduction Using R* (Wiley, Chichester, 2005)
67. M.J. Crawley, *The R Book* (Wiley, Chichester, 2007)

68. S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi, DNA-inspired online behavioral modeling and its application to spambot detection. IEEE Intell. Syst. **31**(5), 58–64 (2016)
69. S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi, Social fingerprinting: detection of spambot groups through DNA-inspired behavioral modeling. IEEE Trans. Dependable Secure Comput. **15**(4), 561–576 (2017)
70. F. Crestani, M. Braschler, J. Savoy, A. Rauber, H. Müller, D.E. Losada, G.H. Bürki, L. Cappellato, N. Ferro, *Experimental IR Meets Multilinguality, Multimodality, and Interaction* (Springer, Cham, 2019)
71. D. Crystal, *The Cambridge Encyclopedia of English Language* (Cambridge University Press, Cambridge, 2003)
72. D. Crystal, *Making Sense of Grammar* (Pearsons, Harlow, 2004)
73. D. Crystal, *'Think on my Words' Exploring Shakespeare's Language* (Cambridge University Press, Cambridge, 2008)
74. D. Crystal, *Txtng: The Gr8 Db8* (Oxford University Press, Oxford, 2008)
75. D. Crystal, *The Cambridge Encyclopedia of Language* (Cambridge University Press, Cambridge, 2010)
76. D. Crystal, *A Little Book of Language* (Yale University Press, Yale, 2010)
77. D. Crystal, *Internet Linguistics* (Routledge, London, 2011)
78. D. Crystal, *Making a Point. The Pernickety Story of English Punctuation* (Profile Books, London, 2016)
79. B. Crystal, D. Crystal, *You Say Potato: The Story of English Accents* (MacMillan, Hampshire, 2015)
80. W. Daelemans, Explanation in computational stylometry, in *Computational Linguistics and Intelligent Text Processing (CICLing)* (Springer, Cham, 2013), pp. 451–462
81. W. Daelemans, M. Kestemont, E. Manjavacas, M. Potthast, F. Rangel, P. Rosso, G. Specht, E. Stamatatos, B. Stein, M. Tschuggnall, M. Wiegmann, E. Zangerle, Overview of PAN 2019: bots and gender profiling, celebrity profiling, cross-domain authorship attribution and style change detection, in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, ed. by F. Crestani, M. Braschler, J. Savoy, A. Rauber, H. Müller, D.E. Losada, G.H. Bürki, L. Cappellato, N. Ferro (Springer, Cham, 2019), pp. 402–416
82. P. Dalgaard, *Introductory Statistics with R* (Springer, Heidelberg, 2002)
83. F. Damereau, The use of function word frequencies as indicator of style. Comput. Humanit. **9**(6), 271–280 (1975)
84. C. Davies, *Divided by a Common Language. A Guide to British and American English* (Houghton Mifflin Harcourt, Boston, 2007)
85. M. De Choudhury, E. Kiciman, M. Dredze, G. Coppersmith, M. Kumar, Discovering shifts to suicidal ideation from mental health content in social media, in *Proceedings Conference on Human Factor in Computing Systems (SIGCHI'16)* (The ACM Press, New York, 2016), pp. 2098–2110
86. A. de Morgan, Letter to Rev. Heald 18/08/1851, in *Memoirs of Augustus de Morgan by his Wife Sophia Elizabeth de Morgan with Selections from his Letters*, ed. by S. Elizabeth, D. Morgan (Longman's Green and Co., London, 1851)
87. M. Del Vicario, A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H.E. Stanley, W. Quattrociocchi, The spreading of misinformation online. Proc. Natl. Acad. Sci. **113**(3), 554–559 (2016)
88. M.P. Deisenroth, A.A. Faisal, C.S. Ong, *Mathematics for Machine Learning* (Cambridge University Press, Cambridge, 2020)
89. L. Deng, J. Wiebe, MPQA 3.0: an entity/event-level sentiment corpus. In *Proceedings Human Language Technologies (HLT/NAACL)* (2015), pp. 1323–1328
90. G. Desagulier, *Corpus Linguistics and Statistics with R* (Springer, Heidelberg, 2017)
91. S.H.H. Ding, B.C.M. Fung, F. Iqbal, W.K. Cheung, Learning stylometric representation for authorship analysis. IEEE Trans. Cybernet. **49**(1), 107–121 (2019)

92. P. Dixon, D. Mannion, Goldsmith's periodical essays: a statistical analysis of eleven doubtful cases. Lit. Linguis. Comput. **8**(1), 1–19 (1993)
93. R. Dror, L. Peled-Cohen, S. Shlomov, R. Reichart, *Statistical Significance Testing for Natural Language Processing* (Morgan & Claypool, San Francisco, 2020)
94. M. Du, N. Liu, X. Hu, Techniques for interpretable machine learning. Commun. ACM **63**(1), 68–77 (2020)
95. T. Dunning, Accurate methods for the statistics of surprise and coincidence. Comput. Linguis. **19**(1), 61–74 (1993)
96. E. Dwoskin, Trump lashes out at social media companies after Twitter labels tweets with fact checks. Washington Post, 144(May. 26th), 2020
97. P. Eckert, S. McConnell-Ginet, *Language and Gender* (Cambridge University Press, Cambridge, 2013)
98. M. Eder, Does size matter? Authorship attribution, small samples, big problem. Digit. Scholarsh. Human. **30**(2), 167–182 (2015)
99. M. Eder, Rolling Delta. Digit. Scholarsh. Humanit. **31**(3), 457–469 (2016)
100. M. Eder, Visualization in stylometry: cluster analysis using networks. Digit. Scholarsh. Humanit. **32**(1), 50–64 (2017)
101. M. Eder, Elena Ferrante: a virtual author, in *Drawing Elena Ferrante's Profile*, ed. by A. Tuzzi, M.A. Cortelazzo (eds.) (Padova University Press, Padova, 2018), pp. 31–46
102. M. Eder, J. Rybicki, Do birds of a feather really flock together, or how to choose test samples for authorship attribution. Lit. Linguis. Comput. **28**(2), 229–236 (2013)
103. M. Eder, J. Rybicki, M. Kestemont, Stylometry with R: a package for computational text analysis. R J. **8**(1), 107–121 (2016)
104. P. Edmondson, S. Wells (eds.), *Shakespeare, Beyond Doubt. Evidence, Argument, Controversy* (Cambridge University Press, Cambridge, 2013)
105. B. Efron, T. Hastie, *Computer Age Statistical Inference. Algorithms, Evidence, and Data Science* (Cambridge University Press, Cambridge, 2016)
106. B. Efron, R. Thisted, Estimating the number of unseen species: How many words did Shakespeare know? Biometrika **63**(3), 435–447 (1976)
107. F.J. Eisenstein, *Introduction to Natural Language Processing* (The MIT Press, Cambridge, 2019)
108. S.E.M. El, I. Kassou, Authorship analysis studies: a survey. Int. J. Comput. Appl. **86**(12), 22–29 (2014)
109. D.Y. Espinosa, H. Gómez-Adorno, G. Sidorov, Bots and gender profiling using character bigrams. Notebook for PAN at CLEF 2019, in *Working Notes Papers of the CLEF 2019 Evaluation Labs Volume 2380 of CEUR Workshop* (CEUR, Aachen, 2019)
110. J. Estepa, Sean Spicer says 'covfefe' wasn't a typo: Trump knew 'exactly what he meant'. USA Today, May 31, 2017
111. S. Evert, T. Proisl, F. Jannidis, I. Reger, S. Pielström, C. Schöch, T. Vitt, Understanding and explaining Delta measures for authorship attribution. Digit. Scholarsh. Humanit. **32**(2), ii4–ii16 (2017)
112. C. Fautsch, J. Savoy, Algorithmic stemmers or morphological analysis? An evaluation. J. Am. Soc. Inf. Sci. **60**(8), 1616–1624 (2009)
113. C. Fellbaum, Wordnet and wordnets, in *Encyclopedia of Language and Linguistics*, ed. by K. Brown (Elsevier, Amsterdam, 2005), pp. 665–670
114. C. Fellbaum, G.A. Miller, *WordNet: An Electronic Lexical Database* (The MIT Press, Cambridge, 1998)
115. E. Ferrara, O. Varol, F. Menczer, A. Flammini, Using sentiment to detect bots on twitter: are humans more opinionated than bots? in *Proceedings of the IEEE/ACM Conference on Advances in Social Networks Analysis and Mining (ASONAM'14)* (2014), pp. 620–627
116. E. Ferrara, O. Varol, F. Menczer, A. Flammini, Detection of promoted social media campaigns, In *Proceedings of the 10th AAAI Conference on Web and Social Media (ICWSM 2016)* (2016), pp. 563–566

117. O. Ferret, Typing relations in distributional thesauri, in *Language Production, Cognition, and the Lexicon*, pp. 113–134 (Springer, Cham, 2014)
118. N. Ferro, What happened in CLEF . . . for a while? in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, ed. by F. Crestani, M. Braschler, J. Savoy, A. Rauber, H. Müller, D. Losada, G. Heinatz, L. Cappellato, N. Ferro (eds.) (Springer, Berlin, 2019)
119. J.R. Firth, A synopsis of linguistic theory 1930–1955, in *Studies in Linguistic Analysis* (Blackwell, Oxford, 1957), pp. 1–32
120. G. Forman, An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. **3**, 1289–1305 (2003)
121. R.S. Forsyth, Stylochronometry with substrings, or: a poet young and old. Lit. Linguis. Comput. **14**(4), 467–478 (1999)
122. O. Fourkioti, S. Symeonidis, A. Arampatis, Language models and fusion for authorship attribution. Inf. Process. Manage. **6**(56), 102061 (2019)
123. W.N. Francis, H. Kucera, *Frequency Analysis of English Usage* (Houghton Mifflin Co., Boston, 1982)
124. G. Fung, O. Mangasarian, The disputed *Federalist Papers*: SVM feature selection via concave minimization, in *Proceedings on Diversity in Computing* (2003), pp. 42–46
125. W.A. Gale, K.W. Church, What is wrong with adding one? in *Corpus-Based Research into Language*, ed. by N. Oostdijk, P. de Hann (Harcourt Brace, New York, 1994)
126. L. Gavalotti, F. Sebastiani, M. Simi, Experiments on the use of feature selection and negative evidence in automated text categorization, in *Proceedings European Conference in Digital Libraries (ECDL)*. Lecture Notes in Computer Science, vol. 1923 (Springer, Heidelberg, 2000), pp. 59–68
127. C. Gelderman, *All the Presidents' Words. The Bully Pulpit and the Creation of the Virtual Presidency* (Walker & Co., New York, 1997)
128. F.A. Gers, J. Schmidhuber, LSTM recurrent networks learn simple context free and context sensitive languages. IEEE Trans. Neural Netw. **12**(6), 1333–1340 (2005)
129. A. Giachanou, J. Gonzalo, F. Crestani, Propagating sentiment signals for estimating reputation polarity. Inf. Process. Manage. **6**(56), 102079 (2019)
130. G. Giodan, C. Saint-Blancat, S. Sbalchiero, Exploring the history of American sociology through topic modelling, in *Tracing the Life Cycle of Ideas in the Humanities and Social Sciences*, ed. by A. Tuzzi (Springer, Cham, 2018), pp. 45–64
131. M. Glickman, J. Brown, Assessing authorship of Beatles songs from musical content: Bayesian classification modeling from bags-of-words representations, in *Proceedings JSM, American Statistical Association* (2018)
132. Y. Goldberg, *Neural Network Methods for Natural Language Processing* (Morgan & Claypool Publishers, San Rafael, 2017)
133. H. Gómez Adorno, A.I. Valencia, C. Stephens Rhodes, G. Fuentes Pineda, Bots and gender identification based on stylometry of tweet minimal structure and n-grams model. Notebook for PAN at CLEF 2019, in *Working Notes Papers of the CLEF 2019 Evaluation Labs volume 2380 of CEUR Workshop* (CEUR, Aachen, 2019)
134. I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning* (The MIT Press, Cambridge, 2016)
135. N. Graham, G. Hirst, B. Marthi, Segmenting documents by stylistic character. Nat. Lang. Eng. **11**(4), 397–415 (2005)
136. A. Granados, M. Cebirán, D. Camacho, F. de Borja Rodríguez, Reducing the loss of information through annealing text distortion. IEEE Trans. Knowl. Data Eng. **23**(7), 1090–1102 (2011)
137. T. Grant, TXT 4N6: method consistency, and distinctiveness in the analysis of SMS messages. J. Law Policy **21**(2) (2012). Article 9
138. C. Gregori-Signes, B. Clavel-Arroitia, Analysing lexical density and lexical diversity in the university students' written discourse, in *Proceedings International Conference on Corpus Linguistics* (2015), pp. 546–556
139. S. Gries, *Quantitative Corpus Linguistics with R: A Practical Introduction* (Routledge, London, 2019)

140. P. Grzybek, E. Kelih, E. Stadlober, The relationship between word length and sentence length: an intra-systemic perspective in the core data structure. Glottometrics **16**, 111–121 (2008)
141. P. Guiraud, *Les caractères statistiques du vocabulaire* (Presses Universitaires de France, Paris, 1954)
142. P. Guiraud, *Essais de stylistique* (Klincksieck, Paris, 1969)
143. S.C. Guntuku, D.B. Yaden, M.L. Kern, L.H. Ungar, J.C. Eichstaedt, Detecting depression and mental illness on social media: an integrative review. Curr. Opin. Behav. Sci. **18**, 43–49 (2017)
144. M. Hagen, M. Potthast, B. Stein, Overview of the author obfuscation task at PAN 2017: safety evaluation revisited, in *Working Notes Papers of the CLEF 2017 Evaluation Labs Volume 1866 of CEUR Workshop*, ed. by L. Cappellato, N. Ferro, L. Goeuriot, T. Mandl (CEUR, Aachen, 2017)
145. A. Hall, L. Terveen, A. Halfaker, Bot detection in Wikipedia using behavioral and other informal cues, in *Proceedings of the ACM on Human-Computer Intercation* (2018), pp. 620–627
146. H.V. Halteren, Author verification by linguistic profiling: An exploration of the parameter space. ACM Trans. Speech Lang. Process. **4**(1) (2007). Article 1
147. O. Halvani, C. Winter, L. Graner, On the usefulness of compression models for authorship verification, in *ARES'17* (The ACM Press, New York, 2017), pp. 1–32
148. O. Halvani, L. Graner, I. Vogel, Authorship verification in the absence of explicit features and thresholds, in *Proceedings European Conference in Information Retrieval (ECIR)*. Lecture Notes in Computer Science, vol. 10772 (Springer, Heidelberg, 2018), pp. 454–465
149. R.A. Hardcastle, CUSUM: a credible method for the determination of authorship? Sci. Just. **37**(2), 129–138 (1997)
150. D. Harman, How effective is suffixing? J. Am. Soc. Inf. Sci. **42**(1), 7–15 (1991)
151. D. Harman, Information retrieval: the early years. Found. Trends Inf. Retr. **13**(5), 425–577 (2019)
152. Z. Harris, Distributional structure. Word **10**(23), 146–162 (1954)
153. R.P. Hart, *Verbal Style and The Presidency. A Computer-Based Analysis* (Academic, Orlando, 1984)
154. R.P. Hart, *Trump and Us: What He Says and Why People Listen* (Cambridge University Press, Cambridge, 2020)
155. R.P. Hart, J.P. Childers, C.J. Lind, *Political Tone. How Leaders Talk and Why* (The Chicago University Press, Chicago, 2013)
156. T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning. Data Mining, Inference, and Prediction* (Springer, New York, 2009)
157. G. Herdan, *Quantitative Linguistics* (Butterworth, London, 1964)
158. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1996)
159. T. Hofmann, Probabilistic latent semantic indexing, in *Proceedings of the International Conference on Information Retrieval (SIGIR 1999)* (The ACM Press, New York, 1999), pp. 50–57
160. D.R. Hoffman, A.D. Howard, *Addressing the State of the Union. The Evolution and Impact of the President's Big Speech* (Lynne Rienner, Boulder, 2006)
161. D.I. Holmes, A stylometric analysis of Mormon scripture and related text. J. R. Stat. Soc. **155**(1), 91–120 (1992)
162. D.I. Holmes, The *Federalist* revisited: new directions in authorship attribution. Lit. Linguis. Comput. **10**(1), 111–127 (1995)
163. D.I. Holmes, The evolution of stylometry in humanities scholarship. Lit. Linguis. Comput. **13**(3), 111–117 (1998)
164. J. Holmes, Woman talk too much, in *Language Myths*, ed. by L. Bauer, P. Trudgill (Penguin Books, London, 1998), pp. 41–49
165. D.I. Holmes, J. Kardos, Who was the author? An introduction to stylometry. Chance **16**(2), 5–8 (2003)

166. D.I. Holmes, F.J. Tweedie, Forensic stylometry: a review of the CUSUM controversy. Revue Informatique et Statistique dans les Sciences Humaines **31**(1), 19–47 (1995)
167. D.L. Hoover, Another perspective on vocabulary richness. Comput. Humanit. **37**(2), 151–178 (2003)
168. D.L. Hoover, Delta prime? Lit. Linguis. Comput. **19**(4), 477–495 (2004)
169. D.L. Hoover, Testing Burrows' Delta. Lit. Linguis. Comput. **19**(4), 453–475 (2004)
170. D.L. Hoover, Teasing out authorship and style with t-tests and Zeta, in *Proceedings Digital Humanities* (2010), pp. 1–3
171. D.L. Hoover, The microanalysis of style variation. Digit. Scholarsh. Humanit. **32**(Supplement 2), ii17–ii30 (2017)
172. D.L. Hoover, S. Hess, An exercise in non-ideal authorship attribution: the mysterious Maria Ward. Lit. Linguis. Comput. **24**(4), 467–489 (2009)
173. P.N. Howard, S. Woolley, R. Calo, Algorithms, bots, and political communication in the US 2016 election: the challenge of automated political communication for election law and administration. J. Inf. Technol. Polit. **15**(2), 81–93 (2018)
174. J. Hudson, S. Mekhennet, G-7 failed to agree on statement after U.S. insisted on calling coronavirus outbreak 'Wuhan virus'. Washington Post, 144, March 25th, 2020
175. J.M. Hughes, N.J. Foti, D.C. Krakauer, D.N. Rockmore, Quantitative patterns of stylistic influence in the evolution of literature. Proc. Natl. Acad. Sci. **109**(20), 7682–7686 (2012)
176. J. Humes, *Confessions of a White House Ghostwriter: Five Presidents and Other Political Adventures* (Regnery Publishing, New York, 1997)
177. C. Ikae, S. Nath, J. Savoy, Unine at PAN-CLEF 2019: Bots and gender task, in *Working Notes Papers of the CLEF 2019 Evaluation Labs volume 2380 of CEUR Workshop* (CEUR, Aachen, 2019)
178. C.R. Jacobsen, M. Nielsen, Stylometry of painting using hidden Markov modelling of contourlet transforms. Signal Process. **93**(3), 579–591 (2013)
179. G. James, D. Witten, T. Hastie, R. Tibshirani, *An Introduction to Statistical Learning with Applications in R* (Springer, New York, 2013)
180. M.L. Jockers, *Macroanalysis. Digital Methods and Literary History* (University of Illinois Press, Urbana, 2013)
181. M.L. Jockers, Testing authorship in the personal writings of Joseph Smith using NSC classification. Lit. Linguis. Comput. **28**(3), 371–381 (2013)
182. M.L. Jockers, *Text Analysis with R for Students of Literature* (Springer, New York, 2014)
183. M.L. Jockers, D.M. Witten, A comparative study of machine learning methods for authorship attribution. Lit. Linguis. Comput. **25**(2), 215–223 (2010)
184. M.L. Jockers, D.M. Witten, C. Criddle, Reassessing authorship of the *Book of Mormon* using Delta and nearest shrunken centroid classification. Lit. Linguis. Comput. **23**(4), 465–491 (2008)
185. V. Johansson, Lexical diversity and lexical density in speech and writing. Working Papers, Lund University, vol. 53, pp. 61–79, 2008
186. F. Johansson, Supervised classification of Twitter accounts based on textual content of tweets. Notebook for PAN at CLEF 2019, in *Working Notes Papers of the CLEF 2019 Evaluation Labs volume 2380 of CEUR Workshop* (CEUR, Aachen, 2019)
187. M. Joos, *The Five Clocks. A Linguistic Excursion into the Five Styles of English Usage* (Harvest/HBJ Book, New York, 1961)
188. P. Joule, D. Vescovi, Analyzing stylometric approaches for author obfuscation, in *Conference on Digital Forensics* (Springer, Berlin, 2011), pp. 115–125
189. P. Juola, The time course of language change. Comput. Humanit. **37**(1), 77–96 (2003)
190. P. Juola, Authorship attribution. Found. Trends Inf. Retr. **1**(3), 233–334 (2006)
191. P. Juola, How a computer program helped show J.K. Rowling write a *Cuckoo's Calling*. Scientific American, August 20th, 2013
192. P. Juola, Using the Google n-gram corpus to measure cultural complexity. Lit. Linguis. Comput. **28**(4), 668–675 (2013)

193. P. Juola, The Rowling case: a proposed standard analytic protocol for authorship questions. Digit. Scholarsh. Humanit. **30**(1), i100–i113 (2016)
194. P. Juola, Thesaurus-based semantics similarity judgments: a new approach to authorship similarity? in *Drawing Elena Ferrante's Profile*, ed. by A. Tuzzi, M.A. Cortelazzo (Padova University Press, Padova, 2018), pp. 47–59
195. P. Juola, G.K. Mikros, S. Vinsick, Correlations and potential cross-linguistic indicators of writing style. J. Quant. Linguis. **26**(2), 146–171 (2019)
196. G. Kacmarcik, M. Gamon, Obfuscating document stylometry to preserve author anonymity, in *Proceedings of the Conference on Computational Linguistics (COLING-ACL)* (The ACL Press, Stroudsburg, 2006), pp. 444–451
197. O.V. Kakushkina, A.A. Polikarpoc, D.V. Khmelev, Using literal and grammatical statistics for authorship attribution. Probl. Inf. Transm. **37**(2), 172–184 (2001)
198. D. Kalb, G. Peters, State of the Union. *Presidential Rhetoric from Woodrow Wilson to George W. Bush* (CQ Press, Washington, 2007)
199. D. Kalb, G. Peters, *Analysis of Phylogenetics and Evolution with R* (Springer, New York, 2012)
200. A. Karpathy, The unreasonable effectiveness of recurrent neural networks, May 2015
201. L. Kaufman, P.J. Rousseeuw, *Finding Groups in Data. An Introduction to Cluster Analysis* (Wiley, Hoboken, 2005)
202. J. Kelleher, *Deep Learning* (The MIT Press, Cambridge, 2019)
203. C. Kesler, C. Rossiter, *The Federalist Papers* (Signet Classic, New York, 2003)
204. M. Kestemont, S. Moens, J. Deploige, Collaborative authorship in the twelfth century: a stylometric study of Hildegard of Birgen and Guibert of Gembloux. Lit. Linguis. Comput. **20**(2), 199–224 (2015)
205. V. Kešelj, F. Peng, N. Cercone, C. Thomas, N-gram-based author profiles for authorship attribution, in *Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING'03* (The ACL Press, Stroudsburg, 2003), pp. 255–264
206. R. Ketcham, *The Anti-Federalist Papers and Constitutional Convention Debates* (Signet Classic, New York, 2003)
207. B. Kjell, Authorship determination using letter pair frequency features with neural network classifier. Lit. Linguis. Comput. **9**(2), 119–124 (1994)
208. M. Kocher, J. Savoy, A simple and efficient algorithm for authorship verification. J. Assoc. Inf. Sci. Technol. **68**(1), 259–269 (2015)
209. M. Kocher, J. Savoy, Distance measures in author profiling. Inf. Process. Manage. **53**(5), 1103–1119 (2017)
210. M. Kocher, J. Savoy, Distributed language representation for authorship attribution. Digit. Scholarsh. Humanit. **33**(2), 425–441 (2018)
211. M. Kocher, J. Savoy, Evaluation of text representation schemes and distance measures for authorship linking. Digit. Scholarsh. Humanit. **34**(1), 189–207 (2019)
212. M. Kolakowski, T.H. Neale, The president's *State of the Union* message: frequently asked questions. *Congressional Research Service* (RS20021), 2006
213. M. Koppel, J. Schler, Exploiting stylistic idiosyncrasies for authorship attribution, in *IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis* (2003), pp. 69–72
214. M. Koppel, S. Seidman, Detecting pseudoepigraphic texts using novel similarity measures. Digit. Scholarsh. Humanit. **33**(1), 72–81 (2018)
215. M. Koppel, Y. Winter, Determining if two documents are by the same author. J. Assoc. Inf. Sci. Technol. **65**(1), 178–187 (2014)
216. M. Koppel, S. Argamon, A.R. Shimoni, Automatically categorizing written texts by author gender. Lit. Linguis. Comput. **17**(4), 401–412 (2002)
217. M. Koppel, N. Akiva, I. Dagan, Feature instability as a criterion for selecting potential style markers. J. Assoc. Inf. Sci. Technol. **57**(11), 1519–1525 (2006)
218. M. Koppel, J. Schler, E. Bonchek-Dokow, Measuring differentiability: unmasking pseudonymous authors. J. Mach. Learn. Res. **8**(6), 1261–1276 (2007)

219. M. Koppel, J. Schler, S. Argamon, Computational methods in authorship attribution. J. Assoc. Inf. Sci. Technol. **60**(1), 9–26 (2009)
220. M. Koppel, J. Schler, S. Argamon, Authorship attribution in the wild. Lang. Resour. Eval. **45**(1), 83–94 (2011)
221. M. Koppel, J. Schler, S. Argamon, Y. Winter, The 'fundamental problem' of authorship attribution. Engl. Stud. **93**(3), 284–291 (2012)
222. D. Kosmajac, V. Kešelj, Twitter user profiling: bot and gender identification, in *Working Notes Papers of the CLEF 2019 Evaluation Labs volume 2380 of CEUR Workshop* (CEUR, Aachen, 2019)
223. S. Kudugunta, E. Ferrara, Deep neural networks for bot detection. Inf. Sci. **467**, 312–322 (2018)
224. N. Laan, Stylometry and methods. the case of Euripides. Lit. Linguis. Comput. **10**(4), 271–278 (1995)
225. D. Labbé, Experiments on authorship attribution by intertextual distance in English. J. Quant. Linguis. **14**(1), 33–80 (2007)
226. D. Labbé, Romain Gary et Emile Ajar. HAL 00279663, 2008
227. D. Labbé, *Si deux et deux font quatre, Molière n'a pas écrit Dom Juan* (Max Milo, Paris, 2009)
228. C. Labbé, D. Labbé, How to measure the meaning of words? Amour in Corneille's work. Lang. Res. Eval. **39**(4), 335–351 (2005)
229. D. Labbé, C. Labbé, A tool for literary studies. Lit. Linguis. Comput. **21**(3), 311–326 (2006)
230. C. Labbé, D. Labbé, Duplicate and fake publications in the scientific literature. Scientometrics **94**(1), 379–396 (2013)
231. C. Labbé, N. Grima, T. Gautier, B. Favier, J.A. Byrne, Semi-automated fact-checking of nucleotide sequence reagents in biomedical research publications: the Seek and Blastn tool. PLoS One **14**(3), e0213266 (2019)
232. G. Lakoff, E. Wehling, *The Little Blue Book: The Essential Guide to Thinking and Talking Democratic* (Free Press, New York, 2012)
233. M. Lalli, F. Tria, V. Loreto, Data-compression approach to authorship attribution, in *Elena Ferrante: A Virtual Author*, ed. by A. Tuzzi, M.A. Cortelazzo (Padova University Press, Padova, 2018), pp. 61–83
234. Q. Le, T. Mikolov, Distributed representations of sentences and documents, in *Proceedings International Conference on Machine Learning*, vol. 32 (2015), pp. II-1188–II-1196
235. L. Lebart, A. Salem, L. Berry, *Exploring Textual Data* (Kluwer, Dordrecht, 1998)
236. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. Nature **521**, 436–444 (2015)
237. G. Ledger, R. Merriam, Shakespeare, Fletcher, and *The Two Noble Kinsmen*. Lit. Linguis. Comput. **9**(3), 235–248 (1994)
238. J.J. Lee, H.Y. Cho, H.R. Park, N-gram-based indexing for Korean text retrieval. Inf. Process. Manage. **35**(4), 427–441 (1999)
239. R.J. Leigh, J. Casson, D. Ewald, A scientific approach to the Shakespeare authorship question. Lit. Rev. **9**(1), 1–13 (2019)
240. O. Levy, Y. Goldberg, Linguistic regularities in sparse and explicit word representations, in *Proceedings Computational Language Learning* (2014), pp. 171–180
241. M. Li, X. Chen, X. Li, B. Ma, P.M.B. Vitanyi, The similarity metric. IEEE Trans. Inf. Theory **50**(12), 3250–3264 (2004)
242. G.J. Lidstone, Note on the general case of the Bayes-Laplace formula for inductive or a posteriori probabilities. Trans. Fac. Actuaries **8**, 182–192 (1920)
243. E.T. Lim, Five trends in presidential rhetoric: an analysis of rhetoric from George Washington to Bill Clinton. Pres. Stud. Q. **32**(2), 328–348 (2002)
244. D.E. Losada, F. Crestani, J. Parapar, Overview of eRisk: Early risk prediction on the internet. in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, ed. by P. Bellot, C. Trabelsi, J. Mothe, F. Murtagh, J.Y. Nie, L. Soulier, E. SanJuan, L. Cappellato, N. Ferro. Lecture Notes in Computer Science, vol. 11018 (Springer, Cham, 2018), pp. 343–361

245. D.E. Losada, F. Crestani, J. Parapar, Overview of eRisk 2019: early risk prediction on the internet, in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, ed. by F. Crestani, M. Braschler, J. Savoy, A. Rauber, H. Müller, D.E. Losada, G.H. Bürki, L. Cappellato, N. Ferro. Lecture Notes in Computer Science, vol. 11696 (Springer, Cham, 2019), pp. 340–357
246. H. Love, *Attributing Authorship: An Introduction* (Cambridge University Press, Cambridge, 2002)
247. K. Luyckx, W. Daelemans, The effect of author set size and data size in authorship attribution. Lit. Linguis. Comput. **26**(1), 35–44 (2011)
248. P. Maier, Ratification. The People Debate the Constitution, 1787–1788. Simon and Schuster Paperbacks, New York, 2010
249. C.D. Manning, H. Schütze, *Foundations of Statistical Natural Language Processing* (The MIT Press, Cambridge, 2000)
250. C.D. Manning, P. Raghavan, H. Schütze, *Introduction to Information Retrieval* (Cambridge University Press, Cambridge, 2008)
251. D. Mannion, P. Dixon, Sentence-length and authorship attribution: the case of Oliver Goldsmith. Lit. Linguis. Comput. **19**(4), 497–508 (2004)
252. M.P. Marcus, B. Santorini, M.A. Marcinkiewicz, Building a large annotated corpus of English: the Penn Treebank. Comput. Linguis. **19**(2), 313–330 (1993)
253. Y. Marton, N. Wu, L. Hellerstein, On compression-based text classification, in *European Conference on Information Retrieval (ECIR)* (Springer, Cham, 2005), pp. 300–314
254. R. Matthews, T. Merriam, Neural computation in stylometry: an application to the works of Shakespeare and Fletcher. Lit. Linguis. Comput. **8**(4), 203–209 (1993)
255. C. McCormick, BERT word embeddings tutorial, May 2019
256. G. McCulloch, *Because Internet. Understanding the New Rules of Language* (Riverhead Books, New York, 2019)
257. P. McNamee, J. Mayfield, Character n-gram tokenization for European language text retrieval. Inf. Retr. J. **7**(1–2), 73–98 (2004)
258. T. Mendenhall, The characteristic curves of composition. Science **214**, 237–249 (1887)
259. R. Merriam, Letter frequency as a discriminator of authors. Notes Queries **41**(4), 467–469 (1994)
260. M.I. Meyerson, *Liberty's Blueprint. How Madison and Hamilton Wrote the Federalist Papers, Defined the Constitution, and Made Democracy Safe for the World* (Basic Books, Philadelphia, 2008)
261. J.-B. Michel, Y.K. Shen, A.P. Aiden, A. Veres, M.K. Gray, The Google Books Team, J.P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M.A. Nowak, E.L. Aiden, Quantitative analysis of culture using millions of digitized books. Science **331**(6014), 176–182 (2011)
262. J. Michell, *Who Wrote Shakespeare* (Thames and Hudson, London, 1999)
263. T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in *Proceedings of Workshop at ICLR 2013* (2013)
264. T. Mikolov, W.T. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in *Proceedings of NAACL HLT 2013* (The ACL Press, Stroudsburg, 2013), pp. 746–751
265. G.K. Mikros, Blended authorship attribution: Unmasking Elena Ferrante. in *Drawing Elena Ferrante's Profile*, ed. by A. Tuzzi, M.A. Cortelazzo (Padova University Press, Padova, 2018), pp. 85–96
266. A. Miranda-Garcia, J. Calle-Martin, Yule's characteristic K revisited. Lang. Res. Eval. **39**(4), 287–294 (2005)
267. A. Miranda-Garcia, J. Calle-Martin, Function words in authorship attribution studies. Lit. Linguis. Comput. **22**(1), 49–66 (2007)
268. A. Miranda-Garcia, J. Calle-Martin, The authorship of the disputed *Federalist Papers* with an annotated corpus. Engl. Stud. **93**(3), 371–390 (2012)
269. T.M. Mitchell, *Machine Learning* (McGraw-Hill, New York, 1997)
270. D. Mitchell, Type-token models: a comparative study. J. Quant. Linguis. **22**, 1–21 (2015)

271. R. Mitton, Spelling checkers, spelling corrections and the misspellings of poor spellers. Inf. Process. Manage. **23**(5), 495–505 (1987)
272. F. Mosteller, D.L. Wallace, Inference in an authorship problem. J. Am. Stat. Assoc. **58**(302), 275–309 (1963)
273. F. Mosteller, D.L. Wallace, *Inference and Disputed Authorship, The Federalist* (Addison-Wesley, Reading, 1964)
274. M. Motta, The dynamics and political implication of anti-intellectualism in the United States. Am. Polit. Res. **46**(3), 465–498 (2018)
275. C. Muller, *Principes et méthodes de statistique lexicale* (Honoré Champion, Paris, 1992)
276. F. Murtagh, P. Legendre, Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? J. Classif. **31**(3), 274–295 (2014)
277. M.J. Narag, M.N. Soriano, Identifying the painter using texture features and machine learning algorithms, in *Proceedings International Conference on Cryptography, Security, and Privacy (ICCSP'19)* (2019), pp. 201–205
278. T. Neal, K. Sundararajan, A. Fatima, Y. Yan, Y. Xiang, D. Woodard, Surveying stylometry techniques and applications. ACM Comput. Surv. **50**(6) (2019). Article 86
279. L. Neidorf, M.S. Krieger, M. Yakubek, P. Chaudhuri, J.P. Dexter, Large-scale quantitative profiling of the Old English verse tradition. Nat. Hum. Behav. **3**, 560–567 (2019)
280. Y. Neuman, *Computational Personality Analysis: Introduction, Practical Applications and Novel Directions* (Springer, Cham, 2016)
281. R.E. Neustadt, *The Accidental President* (Grossman, New York, 1967)
282. R.E. Neustadt, *The Presidential Power and the Modern Presidents. The Politics of Leadership from Roosevelt to Reagan* (Free Press, New York, 1990)
283. J. Noecker, M. Ryan, P. Juola, Psychological profiling through textual analysis. Lit. Linguis. Comput. **28**(3), 382–387 (2013)
284. J.S. Nye, *Presidential Leadership and the Creation of the American Era* (Princeton University Press, Princeton, 2013)
285. M.P. Oakes, M. Farrow, Use of the chi-squared test to examine vocabulary differences in English language corpora representing seven different countries. Lit. Linguis. Comput. **22**(1), 85–99 (2007)
286. K.A. O'Halloran, C. Coffin, *Getting Started. Describing the Grammar of Speech and Writing* (The Open University, Milton Keynes, 2005)
287. C. Olah, Understanding LSTM networks, August 2015
288. W. Oliveira, E. Justino, L.S. Oliveira, Comparing compression models for authorship attribution. Forensic Sci. Int. **228**, 100–104 (2013)
289. J. Olsson, *Forensic Linguistics* (Continuum, London, 2008)
290. J. Olsson, *Word Crime. Solving Crime Through Forensic Linguistics* (Bloomsbury, London, 2009)
291. J. Olsson, *More Wordcrime. Solving Crime Through Forensic Linguistics* (Bloomsbury, London, 2018)
292. B. Pang, L. Lee, Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales, in *Proceedings Association for Computational Linguistics (ACL)*, pp. 115–124 (The ACL Press, Stroudsburg, 2005)
293. R.R. Panko, What we known about spreadsheet errors. J. End User Comput. **10**(2), 51–21 (1998)
294. G. Park, D.B. Yaden, H.A. Schwartz, M.L. Kern, J.C. Eichstaedt, M. Kosinski, D. Stillwell, L.H. Ungar, M.E.P. Seligman, Women are warmer but no less assertive than men: gender and language on Facebook. PLoS One **11**(5), e0155885 (2016)
295. A. Pawłowski, *Séries temporelles en linguistique: Application à l'attribution de textes, Romain Gary et Emile Ajar* (Slatkine, Lausanne, 1996)
296. L. Pearl, M. Steyvers, Detecting authorship deception: a supervised machine learning approach using author writeprints. Lit. Linguis. Comput. **27**(2), 183–196 (2012)

297. C. Peersman, W. Daelemans, L. Van Vaerenbergh, Predicting age and gender in online social networks, in *International Workshop on Search and Mining User-generated Contents (SMUC'11)* (Springer, Cham, 2011), pp. 37–44
298. A. Penas, A. Rodrigo, A single measure to assess nonresponse, in *Proceedings 49th Conference of the Association for Computational Linguistics (ACL)*, pp. 1415–1424 (The ACL Press, Stroudsburg, 2011)
299. J.W. Pennebaker, *The Secret Life of Pronouns. What Our Words Say About Us* (Bloomsbury Press, New York, 2011)
300. J. Pennington, R. Socher, C.D. Manning, GloVe: Global vectors for word representations, in *Proceedings of the Empirical Methods in Natural Language Processing* (2014), pp. 1532–1543
301. S. Pinker, *The Sense of Style* (Penguin Books, London, 2014)
302. P. Plecháč, K. Bobenhausen, B. Hammerich, Versification and authorship attribution. Pilot study on Czech, German, Spanish, and English poetry. Studia Metrica et Poetica **5**(2), 29–54 (2018)
303. I.-I. Popescu, G. Altmann, P. Grzybek, B.D. Jayaram, R. Köhler, V. Krupa, J. Mačutek, R. Pustet, L. Uhlířovà, M.N. Vidya, *Word Frequency Studies* (De Gruyter Mouton, Berlin, 2009)
304. I.-I. Popescu, K.H. Best, G. Altmann, *Unified Modeling of Length in Language* (RAM-Verlag, Lüdenscheid, 2014)
305. M.F. Porter, An algorithm for suffix stripping. Program **14**, 130–137 (1980)
306. N. Potha, E. Stamatatos, Improving author verification based on topic modeling. J. Assoc. Inf. Sci. Technol. **70**(10), 1074–1088 (2019)
307. M. Potthast, A. Barròn-Cedeno, B. Stein, P. Rosso, Cross-language plagiarism detection. Lang. Resour. Eval. **45**(1), 1–18 (2011)
308. M. Potthast, M. Hagen, B. Stein, Author obfuscation: attacking the state of the art in authorship verification, in *Working Notes Papers of the CLEF 2016 Evaluation Labs volume 1609 of CEUR Workshop* (CEUR, Aachen, 2016)
309. M. Potthast, F. Rangel, M. Tschuggnall, E. Stamatatos, P. Rosso, B. Stein, Overview of PAN'17: author identification, author profiling, and author obfuscation, in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, ed. by G. Jones, S. Lawless, J. Gonzalo, L. Kelly, L. Goeuriot, T. Mandl, L. Cappellato, N. Ferro. Lecture Notes in Computer Science, vol. 10456 (Springer, Berlin, 2017), pp. 275–290
310. M. Potthast, F. Schremmer, M. Hagen, B. Stein, Overview of the author obfuscation task at PAN 2018: a new approach to measuring safety, in *Working Notes Papers of the CLEF 2018 Evaluation Labs Volume 2125 of CEUR Workshop* (CEUR, Aachen, 2018)
311. M. Potthast, P. Rosso, E. Stamatatos, B. Stein, A decade of shared tasks in digital text forensics at PAN, in *Proceedings ECIR2019*. Springer Lecture Notes in Computer Science, vol. 11438 (2019), pp. 291–300
312. R. Queneau, *Exercices de style* (Gallimard, Paris, 1947)
313. F. Rangel, P. Rosso, On the impact of emotions on author profiling. Inf. Process. Manage. **52**(1), 73–92 (2016)
314. F. Rangel, P. Rosso, Overview of the 7th author profiling task at PAN 2019: bots and gender profiling in twitter, in *Working Notes Papers of the CLEF 2019 Evaluation Labs Volume 2380 of CEUR Workshop* (CEUR, Aachen, 2019)
315. F. Rangel, P. Rosso, M. Montes y Gómez, M. Potthast, B. Stein, Overview of the 6th author profiling task at PAN 2018: multimodal gender identification in twitter, in *Working Notes Papers of the CLEF 2018 Evaluation Labs Volume 2125 of CEUR Workshop* (CEUR, Aachen, 2018)
316. J.R. Rao, P. Rohatgi, Can pseudonymity really guarantee privacy? in *Proceedings of the 9th USENIX Security Symposium* (USENIX Association, New Orleans, 2000), pp. 85–96
317. T.R. Reddy, B.V. Vardhan, P.V. Reddy, A survey on authorship profiling techniques. Int. J. Appl. Eng. Res. **11**(5), 3092–3102 (2016)

318. W.J. Ridings, S.B. McIver, *Rating the Presidents: A Ranking of U.S. Leaders, from the Great and Honorable to the Dishonest and Incompetent* (Carol Publishing, Secaucus, 1997)
319. P. Rizvi, An improvement to Zeta. Digit. Scholarsh. Humanit. **34**(2), 419–422 (2019)
320. P. Rizvi, The interpretation of the Zeta test results. *Digit. Scholarsh. Humanit.* **34**(2), 401–418 (2019)
321. A. Rocha, W.J. Scheirer, C.W. Forstall, T. Cavalcante, A. Theophilo, B. Shen, A.R.B. Carvalho, E. Stamatatos, Authorship attribution for social media forensics. IEEE Trans. Inf. Forensics Secur. **12**(1), 5–33 (2017)
322. X. Rong, Word2vec parameter learning explained (2016). arXiv.org. arXiv:1411.2738
323. M. Rosen-Zvi, T. Griffiths, T. Steyvers, P. Smyth, The author-topic model for authors and documents, in *Proceedings of the Uncertainty in Artificial Intelligence* (The AUAI Press, Arlington, 2004), pp. 487–494.
324. M. Rosen-Zvi, C. Chemudugunta T. Griffiths, T. Steyvers, P. Smyth, Learning author-topic models from text corpora. ACM Trans. Inf. Syst. **28**(1) (2010). Article 4
325. J. Rudman, The state of authorship attribution studies: some problems and solutions. Comput. Humanit. **31**(4), 351–365 (1998)
326. J. Rudman, Unediting, de-editing, and editing in non-traditional authorship attribution studies: with an emphasis on the canon of Daniel Defoe. Pap. Bibliogr. Soc. Am. **99**(1), 5–36 (2005)
327. J. Rudman, The twelve disputed *Federalist Papers*: a case for collaboration, in *Proceedings Digital Humanities 2012* (2012), pp. 353–356
328. A. Rule, J.P. Cointet, P.S. Bearman, Lexical shifts, substantive changes, and continuity in *State of the Union* discourse, 1790–2014, in *Proceedings National Academy of Sciences*, vol. 112(35) (2015), pp. 10837–10844
329. D. Rumelhart, G. Hinton, R. Williams, Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
330. J. Rybicki, Partners in life, partners in crime? in *Drawing Elena Ferrante's Profile*, ed. by A. Tuzzi, M.A. Cortelazzo (Padova University Press, Padova, 2018), pp. 111–122
331. J. Rybicki, M. Eder, Deeper Delta across genres and languages: do we really need the most frequent words. Lit. Linguis. Comput. **26**(3), 315–321 (2011)
332. J. Rybicki, M. Heydel, The stylistics and stylometry of collaborative translations: Woolf's night and day in Polish. Lit. Linguis. Comput. **28**(4), 708–717 (2013)
333. J. Rybicki, D.L. Hoover, M. Kestemont, Collaborative authorship: Conrad, Ford and rolling Delta. Lit. Linguis. Comput. **29**(3), 422–431 (2014)
334. G. Sampson, *Empirical Linguistics* (Continuum, London, 2001)
335. J. Savoy, Lexical analysis of US political speeches. J. Quant. Linguis. **17**(2), 123–141 (2010)
336. J. Savoy, Authorship attribution based on specific vocabulary. ACM-Trans. Inf. Syst. **30**(2), 170–199 (2012)
337. J. Savoy, Authorship attribution based on a probabilistic topic model. Inf. Process. Manage. **49**(1), 341–354 (2013)
338. J. Savoy, The *Federalist Papers* revisited:a collaborative attribution scheme, in *Proceedings ASIST 2013*, Montreal, November 2013
339. J. Savoy, Comparative evaluation of term selection functions for authorship attribution. Digit. Scholarsh. Humanit. **30**(2), 246–261 (2015)
340. J. Savoy, Text clustering: an application with the *State of the Union* addresses. J. Assoc. Inf. Sci. Technol. **66**(8), 1645–1654 (2015)
341. J. Savoy, Vocabulary growth study: An example with the *State of the Union* addresses. J. Quant. Linguis. **22**(4), 289–310 (2015)
342. J. Savoy, Estimating the probability of an authorship attribution. J. Assoc. Inf. Sci. Technol. **67**(6), 1462–1472 (2016)
343. J. Savoy, Text representation strategies: an example with the *State of the Union* addresses. J. Assoc. Inf. Sci. Technol. **67**(8), 1858–1870 (2016)
344. J. Savoy, Analysis of the style and the rhetoric of the American presidents over two centuries. Glottometrics **38**(1), 55–76 (2017)

345. J. Savoy, Analysis of the style and the rhetoric of the 2016 US presidential primaries. Digit. Scholarsh. Humanit. **33**(1), 143–159 (2018)
346. J. Savoy, Elena Ferrante unmasked. in *Drawing Elena Ferrante's Profile*, ed. by A. Tuzzi, M.A. Cortelazzo (Padova University Press, Padova, 2018), pp. 123–142
347. J. Savoy, Is Starnone really the author behind Ferrante? Digit. Scholarsh. Humanit. **33**(4), 902–918 (2018)
348. J. Savoy, Trump's and Clinton's style and rhetoric during the 2016 presidential election. J. Quant. Linguis. **25**(2), 168–189 (2018)
349. J. Savoy, Authorship of Pauline epistles revisited. J. Assoc. Inf. Sci. Technol. **70**(19), 1089–1097 (2019)
350. N. Schaetti, J. Savoy, Comparison of visualisable evidence-based authorship attribution using reservoir computing and deep learning architecture. Technical Report, University of Neuchatel, 2020
351. H. Schmid, Improvements in part-of-speech tagging with an application to German, in *Proceedings in the ACL SIGDAT-Workshop* (The ACL Press, Stroudsburg, 1995), pp. 47–50
352. S. Schöberlein, Poe or not Poe? A stylometric analysis of Edgar Allan Poe's disputed writings. Digit. Scholarsh. Humanit. **32**(3), 643–759 (2017)
353. H.A. Schwartz, J.C. Eichstaedt, M.L. Kern, L. Dziurzynski, S.M. Ramones, M. Agrawal, A. Shah, M. Kosinski, D. Stillwell, M.E.P. Seligman, L.H. Ungar, Personality, gender, and age in the language of social media: the open-vocabulary approach. PLoS One **8**(9), e73791 (2013)
354. D. Scully, C.E. Brodley, A compression and machine learning: a new perspective on feature space vectors, in *Data Compression Conference (DCC'06)* (The IEEE Press, Piscataway, 2006), pp. 332–341
355. P. Seargeant, *The Emoji Revolution. How Technology Is Shaping the Future of Communication* (Cambridge University Press, Cambridge, 2019)
356. F. Sebastiani, Machine learning in automated text categorization. ACM Comput. Surv. **14**(1), 1–27 (2002)
357. C.J. Shogan, The president's *State of the Union* address: tradition, function, and policy implications. Congressional Research Service (R40132), 2016
358. C.J. Shogan, T.H. Neale, The president's *State of the Union* address: Tradition, function, and policy implications. *Congressional Research Service* (7-5700), 2012
359. K. Shu, H. Liu, *Detecting Fake News on Social Networks* (Morgan & Claypool, San Francisco, 2019)
360. K. Shu, A. Silva, S. Wang, J. Tang, H. Liu, Fake news detection on social media: a data mining perspective. ACM SIGKDD Explorations Newsletter **1**(19), 22–36 (2017)
361. H.S. Sichel, On a distribution law for word frequencies. J. Am. Stat. Assoc. **70**(351), 542–547 (1975)
362. E.H. Simpson, Measurement of diversity. Nature **163**, 688 (1949)
363. R.B. Slatcher, C.K. Chung, J.W. Pennebaker, Winning words: individual differences in linguistic style among U.S. presidential and vice presidential candidates. J. Res. Personal. **41**, 63–75 (2007)
364. F. Smadja, Retrieving collocations from text: Xtract. Comput. Linguis. **19**(1), 143–178 (1993)
365. G. Smith, *The AI Delusion* (Oxford University Press, Oxford, 2018)
366. G. Smith, J. Cordes, *The 9 Pitfalls of Data Science* (Oxford University Press, Oxford, 2019)
367. J.A. Smith, C. Kelly, Stylistic constancy and change across literary corpora: using measures of lexical richness to date works. Comput. Humanit. **36**(4), 411–430 (2002)
368. V. Sotirova, *The Bloomsbury Companion to Stylistics* (Bloomsbury, London, 2016)
369. K. Sparck Jones, A statistical interpretation of term specificity and its application in retrieval. J. Doc. **60**(5), 493–502 (1972)
370. D. Spiegelhalter, *The Art of Statistics. Learning from Data* (Pelican, London, 2019)
371. R. Sproat, *Morphology and Computation* (The MIT Press, Cambridge, 1992)
372. E. Stamatatos, Authorship attribution based on feature set subspacing ensembles. J. Artif. Intell. Tools **15**(5), 823–838 (2006)

373. E. Stamatatos, A survey of modern authorship attribution methods. J. Assoc. Inf. Sci. Technol. **60**(3), 538–556 (2009)
374. E. Stamatatos, Authorship attribution using text distortion, in *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (ACL)* (The ACL Press, Stroudsburg, 2017), pp. 1138–1149
375. E. Stamatatos, N. Fakotakis, G. Kokkinakis, Computer-based authorship attribution without lexical measures. J. Assoc. Inf. Sci. Technol. **35**(1), 193–214 (2001)
376. E. Stamatatos, W. Daelemans, B. Verhoeven, M. Potthast, B. Stein, J. Juola, M.A. Sanchez-Perez, A. Barrón-Cadeno, Overview of the author identification task at PAN 2014, in *Proceeding CLEF-2014, Working Notes*, ed. by L. Cappellato, N. Ferro, M. Halvey, W. Kraaij (CEUR, Aachen, 2014), pp. 877–897
377. E. Stamatatos, M. Tschuggnall, B. Verhoeven, W. Daelemans, G. Specht, B. Stein, M. Potthast, Clustering by authorship within and across documents, in *Notebook Papers of CLEF 2016 Labs and Workshop* (CEUR, Aachen, 2016)
378. C. Stamou, Stylochronometry: Stylistic development, sequence of composition, and relative dating. Lit. Linguis. Comput. **23**(2), 181–199 (2008)
379. B. Stein, N. Lipka, P. Prettenhofer, Intrinsic plagiarism analysis. Lang. Resour. Eval. **45**(1), 63–82 (2011)
380. J.M. Stella, E. Ferrara, M. De Domenico, Bots increase exposure to negative and inflammatory content in online social systems. Proc. Natl. Acad. Sci. **115**(49), 12435–12440 (2018)
381. P.J. Stone, *The General Inquirer: A Computer Approach to Content Analysis*. (The MIT Press, Cambridge, 1966)
382. D.M. Strong, Y.W. Lee, R.Y. Wang, Data quality in context. Commun. ACM **40**(5), 103–110 (1997)
383. L.M. Stuart, S. Tazhibayeva, A.R. Wagoner, J.M. Taylor, On identifying authors with style, in *Proceedings of the 2013 IEEE Conference on Systems, Man, and Cybernetics* (The IEEE Press, Washington, 2013), pp. 3048–3053
384. I. Sutskever, J. Martens, G. Hinton, Generating text with recurrent neural networks, in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (Omnipress, Madison, 2011), pp. 1017–1024
385. I. Sutskever, O. Vinyls, Q.V. Lee, Sequence to sequence learning with neural networks, in *Advanced in Neural Information Processing Systems 27 (NIPS 2014)*, vol. 28 (The IEEE Press, Washington, 2014), pp. 3104–3112
386. M. Taddy, Document classification by inversion of distributed language representations, in *Proceedings Association for Computational Linguistics (ACL)* (The ACL Press, Stroudsburg, 2014), pp. 45–49
387. K. Tanaka-Ishii, S. Aihara, Computational constancy measures of texts - Yule's K and Rényi's entropy. Comput. Linguis. **41**(3), 481–502 (2015)
388. L. Tassinari, *John Florio, The Man who was Shakespeare* (Giano Books, Montreal, 2009)
389. Y.R. Tausczik, J.W. Pennebaker, The psychological meaning of words: LIWC and computerized text analysis methods. J. Lang. Soc. Psychol. **29**(1), 24–54 (2010)
390. G. Taylor, G. Egan, *The New Oxford Shakespeare: Authorship Companion* (Oxford University Press, Oxford, 2017)
391. G. Taylor, R. Loughnane, The life and theatrical interests of Edward de Vere, seventeenth Earl of Oxford, in *Shakespeare, Beyond Doubt. Evidence, Argument, Controversy*, ed. by P. Edmondson, S. Wells (Cambridge University Press, Cambridge, 2013), pp. 39–48
392. G. Taylor, R. Loughnane, The canon and chronology of Shakspeare's works, in *The New Oxford Shakespeare: Authorship Companion*, ed. by G. Taylor, G. Egan (Oxford University Press, Oxford, 2017), pp. 417–603
393. W.J. Teahan, D.J. Harper, Using compression-based languages model for text categorization, in *Language Modeling for Information Retrieval* (Springer, Cham, 2003), pp. 141–165
394. R. Thisted, B. Efron, Did Shakespeare write a newly-discovered poem? Biometrika **4740**(3), 445–455 (1987)

395. F.N. Thomas, M. Turner, *Clear and Simple as the Truth. Writing Classic Prose* (Princeton University Press, Princeton, 2011)
396. J.R.R. Tolkien, *Beowulf*. The monsters and the critics, in *Proceedings of the British Academy* (1936)
397. P. Törnberg, Echo chambers and viral misinformation: Modeling fake news as complex contagion. PLoS One **13**(9), e0203958 (2018)
398. K. Toutanova, D. Klein, C. Manning, Y. Singer, Feature-rich part-of-speech tagging with a cyclic dependency network, in *Proceedings of HLT-NAACL 2003*, pp. 252–259 (The ACL Press, Stroudsburg, 2003)
399. A.W. Trask, *Deep Learning* (Manning, Shelter Island, 2019)
400. M. Trevisani, A. Tuzzi, A portrait of JASA: the history of statistics through analysis of keyword counts in an early scientific journal. Qual. Quant. **49**(3), 1287–1304 (2013)
401. M. Trevisani, A. Tuzzi, Learning the evolution of disciplines from scientific literature: a functional clustering approach to normalized keyword count trajectories. Knowl.-Based Syst. **146**, 129–141 (2018)
402. J. Tuldava, The development of statistical stylistics (a survey). J. Quant. Linguis. **11**(1–2), 141–151 (2004)
403. J. Tulis, *The Rhetorical Presidency* (Princeton University Press, Princeton, 1987)
404. A. Tuzzi, What to put in the bag? Comparing and contrasting procedures for text clustering. Ital. J. Appl. Stat. **22**(1), 77–94 (2010)
405. A. Tuzzi (ed.), *Tracing the Life Cycle of Ideas in the Humanities and Social Sciences* (Springer, Cham, 2018)
406. A. Tuzzi, M. Cortelazzo, *Drawing Elena Ferrante's Profile* (Padova University Press, Padova, 2018)
407. A. Tuzzi, M. Cortelazzo, What is Elena Ferrante? A comparative analysis of a secretive bestselling Italian writer. Digit. Scholarsh. Humanit. **33**(3), 685–702 (2018)
408. A. Tuzzi, M.A. Cortelazzo, It takes many hands to draw Elena Ferrante's profile, in *Drawing Elena Ferrante's Profile*, ed. by A. Tuzzi, M.A. Cortelazzo (Padova University Press, Padova, 2018), pp. 9–30
409. F.J. Tweedie, R.H. Baayen, How variable may a constant be? Measures of lexical richness in perspective. Comput. Humanit. **32**(5), 323–352 (1998)
410. F.J. Tweedie, S. Singh, D.I. Holmes, Neural network applications in stylometry: the *Federalist Papers*. Comput. Humanit. **30**(1), 1–10 (1996)
411. J. Urbano, H. Lima, A. Hanjalic, Statistical significance testing in information retrieval: an empirical analysis of type I, type II and type III errors, in *Proceedings ACM-SIGIR* (The ACM Press, New York, 2019), pp. 505–514
412. R. van der Goot, N. Ljubešić, I. Matroos, M. Nissim, B. Plank, Bleaching text: abstract features for cross-lingual gender prediction, in *Proceedings of the Annual meeting of the Association for Computational Linguistics (ACL)* (The ACL Press, Stroudsburg, 2018), pp. 383–389
413. O. Varol, E. Ferrara, C.A. Davis, F. Menczer, A. Flammini, Online human-bot interactions: detection, estimation, and characterization, in *Proceedings of the 11th AAAI Conference on Web and Social Media (ICWSM 2017)*, pp. 280–289 (2017)
414. T. Veale, M. Cook, *Twitterbots. Making Machines that Make Meaning* (The MIT Press, Cambridge, 2018)
415. B. Vickers, *Shakespeare, Co-author. A Historical Study of Five Collaborative Plays* (Oxford University Press, Oxford, 2002)
416. H. Voorhees, D. Harman, *The TREC Experiment and Evaluation in Information Retrieval* (The MIT University Press, Cambridge, 2005)
417. P. Vossen, *EuroWordNet: a Multilingual Database with Lexical Semantic Networks* (Kluwer, Dordrecht, 1998)
418. A. Vrij, *Detecting Lies and Deceit. Pitfalls and Opportunities* (Wiley, Chichester, 2008)

419. T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, E. Riloff, S. Patwardhan, Opinionfinder: a system for subjectivity analysis, in *Proceedings Empirical Methods for Natural Language Processing (HLT/EMNLP)* (2005), pp. 34–35
420. I.H. Witten, E. Frank, M.A. Hall, *Data Mining. Practical Machine Learning Tools and Techniques* (Morgan Kaufmann, Burlington, 2013)
421. R. Wittgenstein, *Philosophical Investigations* (Basil Blackwell, London, 1953)
422. D.H. Wolpert, The lack of a priori distinctions between learning algorithms. Neural Comput. **8**, 1341–1390 (1996)
423. D.H. Wolpert, The supervised learning no-free-lunch theorems. in *Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications* (2001), pp. 25–42
424. Y. Yang, X. Liu, A re-examination of text categorization methods, in *Proceedings ACM-SIGIR Conference* (The ACM Press, New York, 1999), pp. 42–49
425. Y. Yang, J.O. Pederson, A comparative study of feature selection in text categorization, in *Proceedings International Conference on Machine Learning* (The ACM Press, New York, 1997), pp. 412–420
426. B. Ycart, Alberti's letter counts. Lit. Linguis. Comput. **29**(2), 255–265 (2014)
427. L. Young, S. Soroka, Affective news: the automated coding of sentiment in political texts. Am. Polit. Res. **29**(2), 205–231 (2012)
428. G. Yule, *The Study of Language*, 7th edn. (Cambridge University Press, Cambridge, 2020)
429. E. Zangerle, M. Tschuggnall, G. Specht, B. Stein, M. Potthast, Overview of the style change detection task at PAN 2019, in *Working Notes Papers of the CLEF 2019 Evaluation Labs Volume 2380 of CEUR Workshop* (CEUR, Aachen, 2019)
430. R. Zbib, L. Zhao, D. Karakos, W. Hartmann, J. DeYoung, Z. Huang, Z. Jiang, N. Rivkin, L. Zhang, R. Schwartz, J. Makhoul, Neural-network lexical translation for cross-lingual IR from text and speech, in *Proceedings ACM-SIGIR* (The ACM Press, New York, 2019), pp. 645–654
431. Y. Zhao, J. Zobel, Entropy-based authorship search in large document collection, in *Proceedings ECIR2007*. Springer Lecture Notes in Computer Science, vol. 4425 (2007), pp. 381–392
432. G.K. Zipf, *The Psychology of Language* (Houghton-Mifflin, Boston, 1935)

# Index





J. Savoy, *Machine Learning Methods for Stylometry*,
https://doi.org/10.1007/978-3-030-53360-1